

Maximisation en temps réel de la puissance de sortie
de piles à combustible microbiennes en utilisant
l'optimisation par essais particuliers

par

Frédéric CHASSÉ

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE
M. Sc. A.

MONTREAL, LE 11 JUIN 2018

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Frédéric Chassé, 2018



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

PRÉSENTATION DU JURY
CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

Mme Lyne Woodward, directrice de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Marc Paquet, président du jury
Département de génie de la production automatisée à l'École de technologie supérieure

M. Michel Perrier, membre du jury
Département de génie chimique à l'École Polytechnique de Montréal

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 28 MAI 2018

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens tout d'abord à remercier sincèrement ma directrice de mémoire, Lyne Woodward, ing., Ph. D., qui m'a épaulé tout au long de cette aventure. C'est sans contredit grâce à son soutien et à l'ouverture qu'elle a portée à mon projet que j'ai pu arriver à ces résultats. J'aurai toujours cet accomplissement attaché à mon bagage, et je vous en remercie infiniment.

Je veux aussi exprimer ma reconnaissance envers les membres du jury, qui m'honorent par le temps qu'ils consacrent à évaluer mon travail.

Je remercie l'ÉTS pour son soutien financier, de même que le CRSNG et le FRQNT. Ce sont ces contributions qui aident les étudiants à évoluer dans le domaine de la recherche.

Je remercie également mes collègues du GRÉPCI pour leur support. Je suis spécialement reconnaissant envers Cong-Long Nguyen, qui a dédié énormément de son temps aux piles à combustible microbiennes, et grâce à qui j'ai pu appliquer concrètement mon travail.

Mes remerciements vont aussi à mes collègues de Carré Technologies pour leur soutien pendant ce cheminement. Je salue tout particulièrement Jean-François Roy et Pierre-Alexandre Fournier pour m'avoir accueilli au sein de leur entreprise. Il est toujours captivant d'être dans un environnement passionné, et je suis fier de faire partie du vôtre.

Je ne peux oublier de mentionner mes amis et collègues du Chinook ÉTS, avec qui j'ai surmonté plusieurs défis, et avec qui j'ai partagé maintes expériences enrichissantes.

Bien entendu, je me dois d'exprimer ma gratitude envers ma famille. Votre attitude positive m'a accompagné toute ma vie, et c'est grâce à votre support que j'ai pu suivre mes passions.

Et à Souhaïla Boulahmi, je te remercie pour tes conseils et ton soutien. Je suis fier d'être à tes côtés et infiniment reconnaissant d'avoir la chance de partager chaque instant avec toi.

MAXIMISATION EN TEMPS RÉEL DE LA PUISSANCE DE SORTIE DE PILES À COMBUSTIBLE MICROBIENNES EN UTILISANT L'OPTIMISATION PAR ESSAIMS PARTICULAIRES

Frédéric CHASSÉ

RÉSUMÉ

L'industrie se tourne de plus en plus vers les sources d'énergie alternatives, et les piles à combustible microbiennes (PCM) s'inscrivent dans cette optique, car elles utilisent des microbes pour transformer l'énergie contenue dans la matière organique en énergie électrique.

Les PCM ont cependant une densité de puissance de sortie très faible, en plus d'une dynamique interne très lente, ce qui rend difficile la récolte de leur énergie. En effet, il n'est possible de récolter le maximum d'énergie d'une PCM que si sa charge externe est adaptée à sa résistance interne. De plus, la dynamique interne lente des PCM implique que leur résistance interne peut varier au fil du temps. Ainsi, afin de résoudre ces problèmes, il faut se tourner vers des techniques d'optimisation en temps réel (OTR).

Les algorithmes d'optimisation par essaims particuliers (OEP) et d'optimisation par perturbations et observations (P&O) sont souvent utilisés pour l'OTR de la puissance de systèmes photovoltaïques (PV). Le P&O a aussi été appliqué sur des PCM, mais l'OEP ne l'a pas encore été, bien que cette approche présente plusieurs avantages par rapport au P&O.

Ainsi, la première contribution de ce travail de recherche est une évaluation de la possibilité d'utiliser les algorithmes basés sur l'OEP pour l'OTR de la puissance de sortie de PCM. La deuxième contribution de ce mémoire est la conception d'un nouvel algorithme d'OTR, soit l'optimisation par essaims particuliers parallélisée avec classificateur (OEPPC).

Les résultats de simulation ont démontré que les algorithmes basés sur l'OEP ont beaucoup de potentiel pour l'OTR de la puissance de PCM et qu'ils surpassent les performances du P&O. Les résultats expérimentaux sur des PCM ont quant à eux démontré que, si l'optimum de puissance varie très lentement, il est nécessaire de modifier l'OEP pour l'adapter à un tel contexte. L'OEPPC avec diversité (OEPPCD) a été proposée pour combler ce manque, et des simulations ont démontré que l'OEPPCD performerait mieux que le P&O.

L'algorithme d'OEPPC a aussi été validé expérimentalement sur un système composé de 15 cellules PV. Les résultats expérimentaux ont démontré que l'OEPPC performe mieux que l'OEP et que le P&O, deux algorithmes éprouvés sur les systèmes PV. Ainsi, l'OEPPC a prouvé être un algorithme efficace pouvant être appliqué à différents types de système.

Mots-clés : énergies renouvelables (ÉR), piles à combustible microbiennes (PCM), optimisation en temps réel (OTR), suivi du point de puissance maximal, optimisation par essaims particuliers (OEP), optimisation par perturbations et observations (P&O).

REAL-TIME MAXIMIZATION OF THE POWER OUTPUT OF MICROBIAL FUEL CELLS USING PARTICLE SWARM OPTIMIZATION

Frédéric CHASSÉ

ABSTRACT

The industry is turning more and more to alternative sources of energy, and microbial fuel cells (MFCs) have been getting a lot of attention for many years in that regard, as they use microbes that transform the energy contained in organic matter into electric energy.

However, MFCs have a very low power density, as well as very slow internal dynamics, which makes it difficult to harvest their energy. Indeed, it is only possible to harvest the maximum amount of energy from an MFC if its external load matches its internal resistance. Moreover, the slow dynamics of MFCs imply that their internal resistance may change over time. Real-time optimization (RTO) techniques are thus needed to address these problems.

Particle swarm optimization (PSO) and perturb and observe optimization (P&O) are algorithms which are often used for the RTO of the power of photovoltaic (PV) systems. P&O has also been used on MFCs, whereas PSO hasn't, even though PSO has many advantages over P&O.

Thereby, the first contribution of this work is a study of the usability of PSO-based techniques for the RTO of the power of MFCs. The second contribution is the conception of a new RTO algorithm called parallel particle swarm optimization with classification (PPSOC) applicable on dynamic systems.

Simulations have shown that PSO-based algorithms have a lot of potential for the RTO of the power of MFCs and that they are better than P&O. Experimental results on MFCs have revealed that, if the power optima are slowly changing, then the PSO-based algorithms need to be modified to adapt them to this constraint. PPSOC with diversity (PPSOCD) has been proposed to address this issue, and simulations have shown that PPSOCD would have a better performance than P&O in this context.

The PPSOC algorithm has also been validated experimentally on a system comprised of 15 PV cells. The experimental results have shown that PPSOC has better performances than PSO and P&O, two algorithms that are well-tested on PV systems. As such, PPSOC has been proven to be an effective algorithm that can be applied to different types of systems.

Keywords: renewable energies, microbial fuel cells (MFC), real-time optimization (RTO), maximum power point tracking (MPPT), particle swarm optimization (PSO), perturb and observe optimization (P&O).

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE BIBLIOGRAPHIQUE.....	5
1.1 Introduction.....	5
1.2 Piles à combustible microbiennes.....	5
1.2.1 Principe de fonctionnement	6
1.2.2 Modélisation	8
1.2.3 Conversion et optimisation de l'énergie extraite des piles à combustible microbiennes.....	10
1.3 Optimisation en temps réel	13
1.3.1 Optimisation.....	14
1.3.2 Optimisation par perturbations et observations	17
1.3.3 Optimisation par essais particuliers	19
1.4 Conclusion	24
CHAPITRE 2 ADAPTATION DE L'OPTIMISATION PAR ESSAIS PARTICULAIRES AU PROBLÈME D'OPTIMISATION EN TEMPS RÉEL.....	25
2.1 Introduction.....	25
2.2 Optimisation par essais particuliers séquentielle	25
2.2.1 Saturation et résolution	26
2.2.2 Pbest.....	27
2.2.3 Détection de perturbations et réaction à celles-ci	28
2.2.4 Positions initiales des particules	29
2.2.5 Condition de fin	30
2.3 Optimisation par essais particuliers parallélisée	30
2.4 Optimisation par essais particuliers parallélisée avec classificateur.....	34
2.4.1 Modifications à l'OEPP pour obtenir l'OEPPC.....	37
2.4.1.1 Affinage de l'optimum avec P&O	39
2.4.1.2 Ajout de mémoire	41
2.4.1.3 Classificateur.....	42
2.4.2 Ordinogrammes de l'OEPPC.....	45
2.4.2.1 Sous-algorithmes d'OEPP de l'OEPPC.....	45
2.4.2.2 Sous-algorithme de P&O de l'OEPPC	46
2.4.2.3 Repositionnement des unités de l'OEPPC	48
2.5 Conclusion	49
CHAPITRE 3 OPTIMISATION EN TEMPS RÉEL DE LA PUISSANCE DE SORTIE DE PILES À COMBUSTIBLE MICROBIENNES	51
3.1 Introduction.....	51
3.2 Simulation des approches P&O, OEPS, OEPP, et OEPPC	51

3.2.1	Procédure expérimentale.....	51
3.2.2	Résultats de simulation	54
3.2.2.1	Définition des cas d'étude et des critères de performance.....	54
3.2.2.2	Optimisation d'unités identiques en présence de perturbations.....	55
3.2.2.3	Optimisation d'unités non identiques en présence de perturbations.....	63
3.2.3	Synthèse des résultats	68
3.3	Évaluation expérimentale des algorithmes	69
3.3.1	Procédure expérimentale.....	70
3.3.1.1	Piles à combustible microbiennes.....	70
3.3.1.2	Système électronique d'acquisition et de commande	72
3.3.1.3	Génération de nombres pseudo-aléatoires	75
3.3.1.4	Tests effectués.....	76
3.3.2	Résultats expérimentaux	77
3.3.3	Synthèse des résultats	85
3.4	Conclusion	89
CHAPITRE 4	OPTIMISATION EN TEMPS RÉEL DE LA PUISSANCE DE SORTIE DE CELLULES PHOTOVOLTAÏQUES	93
4.1	Introduction.....	93
4.2	Procédure expérimentale.....	94
4.2.1	Système électronique d'acquisition et de commande	94
4.2.2	Méthode d'acquisition de données et opération des cellules PV	98
4.3	Résultats.....	101
4.3.1	Optimisation de cellules PV sans perturbations.....	101
4.3.2	Optimisation de cellules PV avec perturbations	106
4.4	Conclusion	109
CONCLUSION	111
RECOMMANDATIONS	115
ANNEXE I	IMPLÉMENTATION MATLAB DU MODÈLE D'UNE PILE À COMBUSTIBLE MICROBIENNE DE PINTO ET AL. (2010).....	117
ANNEXE II	ORDINOGRAMME DÉTAILLÉ DU CLASSIFICATEUR DE L'OEPPC.....	121

ANNEXE III	SIMULATION DE L'OPTIMISATION DE 10 PCM NON IDENTIQUES AVEC PERTURBATIONS SUR QUELQUES UNITÉS.....	123
ANNEXE IV	SCHÉMA ÉLECTRIQUE DE LA CARTE D'ACQUISITION ET DE CONTRÔLE	129
ANNEXE V	OPTIMISATION DE TROIS PCM PAR OEPS AVEC MAUVAISE DÉTECTION DE PERTURBATION PAR LES PARTICULES	143
ANNEXE VI	ANALYSE DE L'OEPPC AVEC AJOUT DE DIVERSITÉ POUR COMPENSER LES VARIATIONS TRÈS LENTES DE LA DYNAMIQUE INTERNE DES PCM	145
ANNEXE VII	SCHÉMA ÉLECTRIQUE DU PCB DE CELLULES PV	151
ANNEXE VIII	SCHÉMA ÉLECTRIQUE DU CIRCUIT DE DEL POUR LES CELLULES PV	153
ANNEXE IX	DESSINS MÉCANIQUES DU BOÎTIER DES CELLULES PV	157
	LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES.....	161

LISTE DES TABLEAUX

	Page
Tableau 3.1 Paramètres de P&O utilisés pour les simulations	56
Tableau 3.2 Paramètres de l'OEPS utilisés pour les simulations.....	57
Tableau 3.3 Paramètres de l'OEPP utilisés pour les simulations.....	59
Tableau 3.4 Paramètres de l'OEPPC utilisés pour les simulations	60
Tableau 3.5 Résultats de simulation de PCM identiques en présence de perturbations	62
Tableau 3.6 Résultats de simulation de PCM non identiques en présence de perturbations	67
Tableau 3.7 Paramètres de P&O pour les tests expérimentaux sur des PCM.....	77
Tableau 3.8 Paramètres d'OEPS et d'OEPP pour les tests expérimentaux sur des PCM	77
Tableau 3.9 Évolution des particules de l'OEPS sur la PCM 2	82
Tableau 3.10 Énergie récoltée par les algorithmes lors des tests sur des PCM.....	86
Tableau 3.11 Temps de convergence des algorithmes lors des tests sur des PCM	86
Tableau 3.12 Puissances finales obtenues par les algorithmes lors des tests sur des PCM	86
Tableau 4.1 Paramètres de P&O utilisés pour les expérimentations	102
Tableau 4.2 Paramètres de l'OEPS utilisés pour les expérimentations.....	102
Tableau 4.3 Paramètres de l'OEPPC utilisés pour les expérimentations	103
Tableau 4.4 Résultats de l'optimisation de 15 cellules PV sans perturbations	105
Tableau 4.5 Résultats de l'optimisation de 15 cellules PV avec perturbations.....	109

LISTE DES FIGURES

	Page
Figure 1.1	Diagramme d'une PCM à deux compartiments8
Figure 1.2	Exemple de PCM à compartiment unique8
Figure 1.3	Représentation graphique du modèle de Pinto et al. (2010).....9
Figure 1.4	Puissance de sortie d'une source de tension avec une résistance interne en fonction de sa charge externe.....12
Figure 1.5	Exemple d'une solution au problème du voyageur de commerce15
Figure 1.6	Application du P&O sur une fonction objectif statique.....19
Figure 1.7	Ordinogramme de l'optimisation par essais particuliers21
Figure 1.8	Simulation de l'OEP sur une fonction objectif statique22
Figure 2.1	Ordinogrammes de a) l'OEP séquentielle, et de b) l'OEP Parallélisée32
Figure 2.2	Différences statiques entre deux unités.....34
Figure 2.3	Étapes par lesquelles une unité optimisée par OEPPC peut passer37
Figure 2.4	Ordinogramme de l'OEPPC39
Figure 2.5	Exemple de classification de 10 unités dans un espace de recherche D43
Figure 2.6	Ordinogramme du classificateur44
Figure 2.7	Ordinogramme de l'OEPP de l'OEPPC46
Figure 2.8	Ordinogramme du P&O de l'OEPPC47
Figure 2.9	Ordinogramme de l'algorithme de repositionnement des unités.....48

Figure 3.1	Exemple graphique de l'implémentation du modèle d'une PCM de Pinto et al. (2010)	52
Figure 3.2	Courbes de puissance statiques de deux PCM ayant un S_0 différent	54
Figure 3.3	Optimisation par P&O de PCM identiques en présence de perturbations.....	56
Figure 3.4	Optimisation par OEPS de PCM identiques en présence de perturbations.....	58
Figure 3.5	Optimisation par OEPP de PCM identiques en présence de perturbations.....	59
Figure 3.6	Optimisation par OEPPC de PCM identiques en présence de perturbations	61
Figure 3.7	Courbes statiques de puissance de PCM à différents S_0	63
Figure 3.8	Optimisation par P&O de PCM non identiques en présence de perturbations	64
Figure 3.9	Optimisation par OEPS de PCM non identiques en présence de perturbations	65
Figure 3.10	Optimisation par OEPP de PCM non identiques en présence de perturbations	65
Figure 3.11	Optimisation par OEPPC de PCM non identiques en présence de perturbations	66
Figure 3.12	Modèle 3D du boîtier d'une PCM	71
Figure 3.13	Montage d'une PCM.....	71
Figure 3.14	Courbes statiques de puissance des PCM	72
Figure 3.15	Carte électronique utilisée dans le banc d'essai	73
Figure 3.16	Représentation du branchement d'une PCM au circuit de contrôle et d'acquisition	74
Figure 3.17	Données brutes de l'optimisation de 3 PCM par OEPS	78
Figure 3.18	Données algorithmiques de l'optimisation de 3 PCM par OEPS	79

Figure 3.19	Analyse de la PCM 2 lorsqu'optimisée par OEPS	79
Figure 3.20	Évolution des niveaux de tension des 3 PCM optimisées par OEPS.....	80
Figure 3.21	Données brutes de l'optimisation de 3 PCM par OEPP	83
Figure 3.22	Données algorithmiques de l'optimisation de 3 PCM par OEPP	83
Figure 3.23	Données brutes de l'optimisation de 3 PCM par P&O	84
Figure 3.24	Données algorithmiques de l'optimisation de 3 PCM par P&O	85
Figure 3.25	Exemple d'optimisation par OEPPC avec ajout de diversité de 10 PCM non identiques avec perturbations lentes	89
Figure 4.1	Boîtier du banc d'essai	95
Figure 4.2	Modèle 3D de l'assemblage du boîtier, du PCB de cellules PV, et du mur de séparation	95
Figure 4.3	PCB de cellules PV encapsulé dans le boîtier	96
Figure 4.4	Vue 3D du dessus du PCB de DEL	96
Figure 4.5	Vue du dessous du PCB de DEL du banc d'essai	96
Figure 4.6	Montage complet du banc d'essai	97
Figure 4.7	Dynamique d'une cellule PV.....	98
Figure 4.8	Tension de sortie d'une cellule PV avec un cycle de fonctionnement de 30%	99
Figure 4.9	Tension de sortie d'une cellule PV avec un cycle de fonctionnement de 50%	99
Figure 4.10	Courbes statiques de puissance de 15 cellules PV ayant une intensité lumineuse de 50%.....	102
Figure 4.11	Optimisation par P&O de 15 cellules PV sans perturbations.....	103

Figure 4.12	Optimisation par OEPS de 15 cellules PV sans perturbations.....	104
Figure 4.13	Optimisation par OEPPC de 15 cellules PV sans perturbations.....	104
Figure 4.14	Courbes statiques de puissance de 15 cellules PV ayant une intensité lumineuse de 30%	107
Figure 4.15	Optimisation par P&O de 15 cellules PV avec perturbations sur toutes les unités	107
Figure 4.16	Optimisation par OEPS de 15 cellules PV avec perturbations sur toutes les unités	108
Figure 4.17	Optimisation par OEPPC de 15 cellules PV avec perturbations sur toutes les unités	108

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ADC	<i>Analog to Digital Converter</i> Convertisseur analogique à numérique
CC	Courant Continu
CC/CC	Convertisseur CC à CC, ou alimentation à découpage
DEL	Diode Électroluminescente
ÉR	Énergies Renouvelables
I ² C	<i>Inter-Integrated Circuit</i> Protocole de communication I ² C
MFC	<i>Microbial Fuel Cell</i> Pile à combustible microbienne
MOSFET	<i>Metal Oxide Semiconductor Field Effect Transistor</i> Transistor à effet de champ à grille isolée
MPPT	<i>Maximum Power Point Tracking</i> Suivi du point de puissance maximal
NPA	Nombre Pseudo-Aléatoire
OCV	<i>Open-Circuit Voltage</i> Tension en circuit ouvert
OEP	Optimisation par Essaims Particulaires
OEPP	Optimisation par Essaims Particulaires Parallélisée
OEPPC	Optimisation par Essaims Particulaires Parallélisée avec Classificateur
OEPPCD	Optimisation par Essaims Particulaires Parallélisée avec Classificateur et avec Diversité
OEPS	Optimisation par Essaims Particulaires Séquentielle
OTR	Optimisation en Temps Réel

P&O	<i>Perturb and Observe</i> Perturbations et observations
PCB	<i>Printed Circuit Board</i> Circuit imprimé
PCG	<i>Permuted Congruential Generator</i> Générateur congruent permuté
PCM	Pile à Combustible Microbienne
PPSO	<i>Parallel Particle Swarm Optimization</i> Optimisation par essaims particuliers parallélisée
PPSOC	<i>Parallel Particle Swarm Optimization with Classification</i> Optimisation par essaims particuliers parallélisée avec classificateur
PPSOCD	<i>Parallel Particle Swarm Optimization with Classification and Diversity</i> Optimisation par essaims particuliers parallélisée avec classificateur et avec diversité
PSO	<i>Particle Swarm Optimization</i> Optimisation par essaims particuliers
PV	Photovoltaïque <i>Cellules photovoltaïques, ou système photovoltaïque</i>
PWM	<i>Pulse Width Modulation</i> Modulation de largeur d'impulsion
SA	Sous-Algorithme Sous-algorithme de l'OEPPC
SAR ADC	<i>Successive Approximation Register ADC</i> Convertisseur numérique à analogique à approximations successives
SPI	<i>Serial Peripheral Interface bus</i> Protocole de communication SPI
SPSO	<i>Sequential Particle Swarm Optimization</i> Optimisation par essaims particuliers séquentielle
UART	<i>Universal Asynchronous Receiver-Transmitter</i> Protocole de communication UART

USB *Universal Serial Bus*
Protocole de communication USB

LISTE DES SYMBOLES ET UNITÉS DE MESURE

Δ	Paramètre Δ du P&O
Δ_1	Paramètre Δ_1 de l'OEPPC
Δ_2	Paramètre Δ_2 de l'OEPPC
Δ_c	Marge du classificateur de l'OEPPC
c_1	Coefficient cognitif de l'OEP
c_2	Coefficient social de l'OEP
D_{opt}	Meilleure position d'une unité de l'OEPPC depuis sa dernière perturbation
E_{total}	Énergie récoltée par un algorithme pendant une optimisation
ω	Coefficient d'inertie de l'OEP
d_{opt}	Position optimale relative
IL	Intensité lumineuse reçue par les cellules PV pendant les tests (%)
I_{sc}	Courant de court-circuit (A)
OCV	Tension en circuit ouvert (V)
P_{conv}	Puissance moyenne obtenue à la convergence d'un algorithme (mW)
P_{opt}	Puissance optimale (W)
P_{out}	Puissance de sortie (W)
R_{ext}	Résistance externe (Ω)
R_{int}	Résistance interne (Ω)
R_{opt}	Résistance optimale (Ω)
S_0	Concentration massique du substrat d'alimentation d'une PCM (mg/L)
T_{conv}	Temps de convergence d'un algorithme (itérations)
$T_{diversité}$	Nombre d'itérations après lesquelles on ajoute de la diversité à l'OEPPCD

INTRODUCTION

Les énergies renouvelables (ÉR) sont, depuis plusieurs années, un sujet faisant l'objet de plus en plus d'études. En effet, dans un contexte où les combustibles fossiles représentent les principales sources d'énergie de la société (Logan, 2008), il existe un besoin urgent de trouver des sources d'énergie alternatives, car les ressources disponibles seront épuisées dans un avenir relativement proche. Les ÉR sont propices pour combler un tel besoin étant donné que les ressources utilisées se renouvellent naturellement assez rapidement pour qu'elles soient considérées virtuellement inépuisables. Parmi ces ressources naturelles, on compte notamment l'énergie solaire, l'énergie éolienne, l'énergie hydraulique, l'énergie géothermique, et les biomasses.

Les piles à combustible microbiennes (PCM) sont considérées comme un type de source d'énergie renouvelable, car elles peuvent transformer de l'énergie biologique en énergie électrique. Pour y arriver, elles consomment des matières organiques, tels que des eaux usées, des résidus de plantes, ou autres, et génèrent de l'électricité et de l'eau propre dans le processus (Du, Li et Gu, 2007). Ainsi, les PCM sont désirables en tant qu'ÉR car elles pourraient, entre autres, être utilisées pour transformer des déchets organiques en énergie électrique.

Comme toute technologie, les différents types d'ÉR ont tous des problèmes distincts que plusieurs études tentent de résoudre. Il existe cependant une problématique commune à tous ces systèmes : les ressources utilisées pour extraire l'énergie se retrouvent dans des conditions environnementales propices aux changements, ce qui peut causer des difficultés lors de la récolte d'énergie. Par exemple, dans un système à énergie solaire, où des cellules photovoltaïques (PV) sont utilisées pour transformer l'énergie solaire en énergie électrique, un nuage pourrait bloquer brièvement le soleil, entraînant une baisse de la puissance générée par les cellules PV.

Afin d'amortir les pertes d'énergie pouvant survenir lors des perturbations environnementales, différents algorithmes d'optimisation en temps réel ont été développés. On peut grouper plusieurs de ces algorithmes sous une famille dénommée méthodes de suivi du point de puissance maximal (« maximum power point tracking » en anglais, ou MPPT). Ces méthodes visent à suivre en temps réel l'optimum de puissance d'un système, et ce afin d'adapter ce dernier aux perturbations pouvant affecter le niveau de puissance fournie à tout moment. Dans un système où des perturbations pourraient être fréquentes, la vitesse de convergence de l'algorithme choisi est critique afin d'extraire le maximum d'énergie possible d'un système donné.

Cette exigence des algorithmes d'optimisation en temps réel est d'autant plus vraie pour la PCM car, étant un système bioélectrique, le temps de convergence d'un point d'opération à un autre peut être de l'ordre de plusieurs minutes (Woodward et al., 2009), augmentant ainsi la criticité d'utiliser un algorithme capable de converger rapidement. En outre, les PCM présentent une densité de puissance de sortie très faible (Woodward et al., 2010), donc la capacité de l'algorithme à trouver le point optimal de puissance de façon précise devient aussi un critère majeur du choix de la méthode d'optimisation.

Les algorithmes basés sur l'optimisation par essais particuliers (OEP) sont intéressants dans ce sens. En effet, plusieurs articles scientifiques étudient les performances de l'OEP lorsqu'appliquée sur un système photovoltaïque, et il a été prouvé que ce type d'algorithme est précis et qu'il s'adapte bien aux perturbations externes tout en ayant une vitesse de convergence rapide. Bien que ce type d'algorithme soit éprouvé sur les systèmes PV, l'OEP n'a cependant pas encore été appliquée sur des piles microbiennes.

Ce mémoire a donc deux buts : évaluer les performances d'algorithmes basés sur l'OEP lorsqu'ils sont utilisés comme MPPT sur un système composé de plusieurs PCM, et proposer un algorithme d'optimisation en temps réel pouvant être appliqué sur des piles à combustible microbiennes pour en optimiser la puissance de sortie. Afin d'avoir une simplicité d'implémentation, l'algorithme proposé ne requiert aucun modèle fondamental, mais prend

tout de même avantage des connaissances acquises lors de l'optimisation pour améliorer sa précision et sa vitesse de convergence. L'algorithme proposé est basé sur l'optimisation par essais particuliers (OEP) en raison de la précision de l'algorithme et de sa capacité d'adaptation aux perturbations externes. L'algorithme tente d'améliorer la vitesse de convergence de l'OEP tout en conservant la précision que cette méthode offre. Les performances de l'OEP et de l'algorithme proposé sont comparées à celle de l'optimisation par perturbations et observations (P&O), car cette dernière a souvent été appliquée sur les systèmes PV, et a aussi été utilisée comme MPPT sur des PCM. Deux cas d'étude sont présentés, soient : l'optimisation en temps réel de la puissance de sortie de piles à combustible microbiennes; et l'optimisation en temps réel de la puissance de sortie de cellules photovoltaïques. Le cas d'étude avec des PCM est d'abord fait par simulations, puis sur un banc d'essai en laboratoire, et le cas d'étude avec des cellules photovoltaïques est fait uniquement sur un banc d'essai en laboratoire.

Le mémoire est divisé comme suit. Le CHAPITRE 1 présente un survol du principe de fonctionnement d'une PCM, ainsi qu'une étude des algorithmes P&O et OEP, deux techniques d'optimisation grandement utilisées dans le domaine des énergies renouvelables. Le CHAPITRE 2 élabore différents algorithmes qui sont étudiés dans ce mémoire. Ces algorithmes sont testés par simulations et expérimentations sur des PCM au CHAPITRE 3. Le CHAPITRE 4 donne les résultats d'expérimentations de ces algorithmes sur un système composé de cellules photovoltaïques. Une conclusion faisant la synthèse du mémoire et offrant diverses avenues possibles pour de futures études est ensuite faite.

CHAPITRE 1

REVUE BIBLIOGRAPHIQUE

1.1 Introduction

Ce chapitre explique tout d'abord de manière succincte le fonctionnement des PCM et les défis qu'elles représentent. Ensuite, deux techniques d'optimisation en temps réel sont présentées, de même que la façon dont elles peuvent être appliquées pour optimiser la puissance de sortie des PCM. Ces algorithmes sont l'optimisation par perturbations et observations (P&O), et l'optimisation par essais particuliers (OEP). Une attention particulière est portée à l'OEP, car les algorithmes étudiés dans ce mémoire sont basés sur cette approche. L'optimisation par P&O est, quant à elle, présentée car elle sert de base de comparaison lors de l'évaluation des performances des algorithmes étant donné sa simplicité et le fait qu'elle est utilisée dans un grand nombre d'études.

1.2 Piles à combustible microbiennes

Les premières observations d'électricité générée par des bactéries datent de plus d'un siècle et sont attribuées à Potter (Potter, 1911), mais peu d'études furent produites dans les années qui suivirent. Un intérêt pour les PCM est apparu dans les années 1990 (Logan, 2008) et, depuis, plusieurs recherches ont été faites dans le but de mieux comprendre cette technologie et les diverses façons de l'exploiter (Du, Li et Gu, 2007), (Li, Yu et He, 2014).

Étant une technologie relativement récente, les PCM n'ont pas encore été adaptées aux besoins de l'industrie à leur pleine capacité. Cependant, plusieurs applications possibles pour les PCM ont été proposées. Notamment, Logan (2008) indique que, pour l'instant, le domaine du traitement des eaux usées est celui où le potentiel des PCM est le plus haut. En effet, au lieu d'utiliser de l'énergie électrique pour nettoyer les eaux usées, les PCM pourraient générer de l'électricité tout en diminuant de 50 à 90% la quantité de solides indésirables dans l'eau municipale (Du, Li et Gu, 2007). En outre, Li, Yu et He (2014) ont

proposé une approche qui intégrerait les PCM dans des processus de traitement des eaux usées déjà existants. Selon eux, la stratégie qu'ils proposent constitue une bonne base vers l'implantation efficace et durable des PCM dans les usines de traitement des eaux usées.

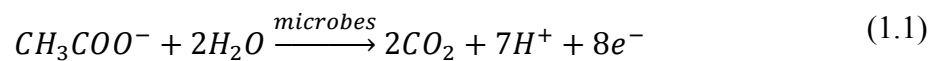
La génération d'énergie électrique pour alimenter des systèmes électroniques pourrait aussi être une application possible des PCM. Du, Li et Gu (2007) parlent d'EcoBot II, un robot alimenté entièrement avec des PCM (Melhuish et al., 2006) auxquelles sont fournies des sucres, fruits, insectes et herbes récoltées par le robot. Du, Li et Gu proposent aussi d'utiliser les PCM dans des navettes spatiales, où elles pourraient faire la biodégradation de déchets biologiques générés à bord pour produire de l'électricité. Shantaram et al. (2005) ont réussi à alimenter des capteurs électrochimiques et un transmetteur avec une PCM branchée sur un convertisseur électrique ayant une haute efficacité (Shantaram et al., 2005). Cette étude indique que, bien que les PCM aient peut-être plus de potentiel dans le secteur du traitement des eaux usées, elles peuvent aussi être utilisées pour alimenter des circuits électroniques ayant une faible consommation de puissance.

Plusieurs autres applications des PCM sont possibles et des recherches se font pour en trouver d'autres. Le défi reste cependant le même, soit d'améliorer la puissance que peuvent fournir des PCM.

1.2.1 Principe de fonctionnement

Les PCM sont des piles utilisant des microbes exogènes comme catalyseurs pouvant générer de l'électricité en oxydant une biomasse (Logan, 2008). Les équations (1.1) et (1.2), tirées de Du, Li et Gu, 2007, représentent les réactions typiques qui se produisent à l'anode et à la cathode de la pile, respectivement. Ainsi, en fournissant de l'eau usée ou toute autre source de carbone aux piles comme combustible (par exemple, l'équation (1.1) utilise de l'acétate (CH_3COO^-)), on peut générer de l'électricité et de l'eau propre.

Les PCM peuvent être comparées à des piles à combustible normales, c'est-à-dire qu'il y a une réaction d'oxydation de combustible à l'anode et une réaction de réduction d'oxygène à la cathode qui, à deux, génèrent de l'électricité (Carrette, Friedrich et Stimming, 2001). Cependant, au lieu d'utiliser de l'hydrogène ou du méthane comme carburant, une biomasse est fournie à l'anode et est oxydée par les bactéries anodophiles contenues dans la chambre anodique de la pile.



Tirées de Du, Li et Gu (2007)

Les équations (1.1) et (1.2) résument bien les réactions anodiques et cathodiques possibles d'une PCM. Une biomasse est fournie comme substrat à la chambre anodique de la pile. Les microbes oxydent cette biomasse en produisant du CO_2 et génèrent des électrons et des protons. Les protons sont transférés à la chambre cathodique et se mélangent à l'oxygène pour produire de l'eau propre. Dans quelques cas, les protons seront transférés à la cathode au travers d'une membrane séparant les deux chambres. Cependant, les PCM dites « sans membranes » sont plus populaires car elles sont moins coûteuses et plus simples à concevoir. Les électrons, quant à eux, sont ensuite transportés à la chambre cathodique par le biais d'un circuit externe. Ces électrons, voyageant à travers la charge externe, produisent un courant que l'on cherchera à exploiter. La Figure 1.1 présente un diagramme d'une PCM à deux compartiments.

Il est à noter que la chambre anodique doit obligatoirement être anaérobique, car la présence d'oxygène à l'anode empêcherait la génération d'électrons (Logan, 2008). Pour éviter qu'une telle situation se produise, on utilisera souvent une membrane d'échange de protons séparant l'anode et la cathode, créant ainsi les chambres anodique et cathodique, d'où le terme « PCM à deux compartiments ».

On trouve aussi des PCM à compartiment unique. Dans cette topologie, la cathode est directement exposée à l'air libre, ce qui diminue la difficulté de conception de pile. La Figure 1.2 présente un exemple de ce type de PCM.

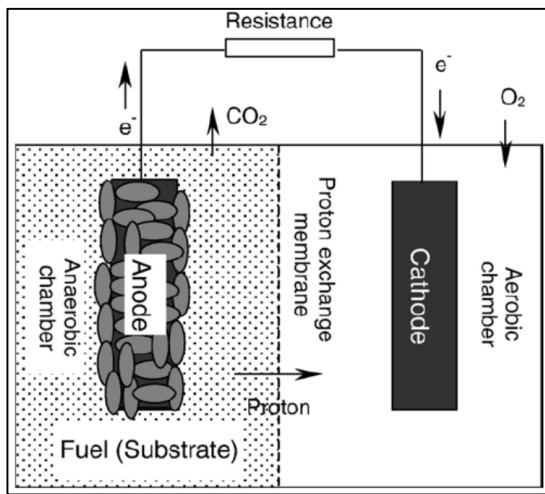


Figure 1.1 Diagramme d'une PCM à deux compartiments
Tirée de Du, Li et Gu (2007)

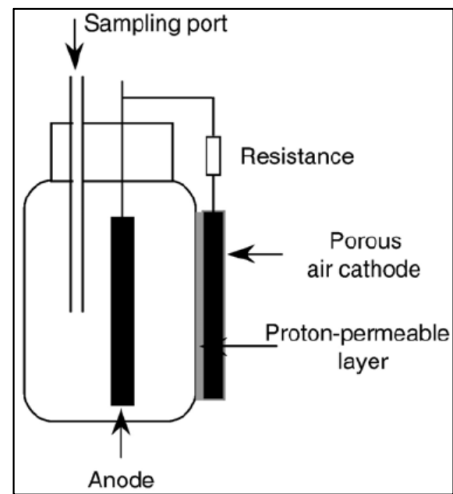


Figure 1.2 Exemple de PCM à compartiment unique
Tirée de Du, Li et Gu (2007)

1.2.2 Modélisation

L'utilisation d'un modèle mathématique peut être pratique lorsque l'on cherche à optimiser un système. En effet, le modèle peut être utilisé pour faire des simulations préalables aux essais expérimentaux. Dans le cas des PCM, des suites de tests expérimentaux peuvent prendre de quelques heures à quelques semaines en raison de leur dynamique interne très lente. Ainsi, en appliquant la méthode d'optimisation choisie au modèle mathématique et en simulant le tout, on peut sauver un temps précieux durant les premières phases de validation d'une stratégie de commande et d'optimisation.

Quelques modèles de PCM ont été élaborés par plusieurs chercheurs (Pinto et al., 2010). Notamment, Pinto et al. (2010) ont développé un modèle d'une PCM axé sur des populations microbiennes anodophiles et méthanogènes. La Figure 1.3 est une représentation graphique du modèle proposé par ces chercheurs.

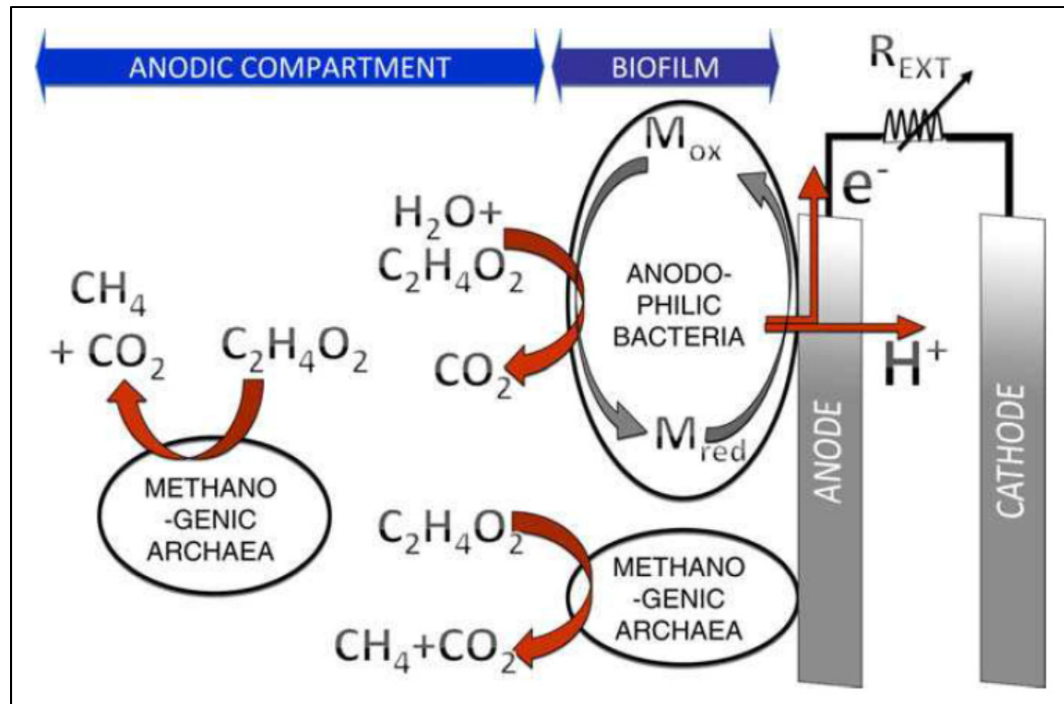


Figure 1.3 Représentation graphique du modèle de Pinto et al. (2010)
Tirée de Pinto et al. (2010)

Le modèle de Pinto et al. est constitué d'équations différentielles décrivant les différents comportements intrinsèques d'une PCM. Les chercheurs séparent les équations en différentes catégories, soient : équations de bilan de matière de la chambre anodique, équations des bilans intracellulaires, équations cinétiques, et équations électrochimiques. Les différentes équations sont bien décrites dans leur article et peuvent être codées directement dans un logiciel de simulation tel que MATLAB/Simulink.

Un modèle plus récent (2016) a été élaboré par l'équipe de recherche de Michel Perrier et de Boris Tartakovsky (Recio-Garrido, Perrier et Tartakovsky, 2016). Ce modèle est plus complet que celui de Pinto et al. car, en plus du modèle électrochimique, un modèle électrique est incorporé. L'inclusion du modèle électrique est faite afin de prendre en compte les différentes échelles de temps pour la dynamique de la pile, i.e. plus lente pour l'aspect électrochimique, et plus rapide pour l'aspect électrique. Cependant, le modèle utilisé dans ce mémoire est celui de Pinto et al., étant donné que les scripts de simulation ont été développés

avant la publication de l'article de Recio-Garrido, Perrier et Tartakovsky. L'ANNEXE I présente l'implémentation MATLAB du modèle de Pinto et al., tel qu'utilisé dans le cadre de cette recherche.

1.2.3 Conversion et optimisation de l'énergie extraite des piles à combustible microbiennes

Tout d'abord, il est à noter que, lorsque l'on parle de la puissance d'une PCM, il est plus sensé de décrire la puissance normalisée au volume d'une pile, ou puissance volumétrique (W/m^3) (Logan, 2008), car les tailles et formes des PCM peuvent grandement varier d'une pile à l'autre. Ainsi, la puissance volumétrique d'une PCM se calculera selon l'équation (1.3), où v représente le volume total du réacteur.

$$P_v = \frac{E_{MFC}^2}{vR_{ext}} \quad (1.3)$$

Tirée de Logan (2008)

Les PCM présentent une densité de puissance de sortie assez faible avec généralement une tension de sortie de 0.3 – 0.5 V (Woodward et al., 2010), mais quelques cas ont observé une tension de 0.8 V (Wang, Park et Ren, 2015). On peut calculer la tension idéale d'une pile grâce à l'équation de Nernst (équation (1.4)), qui décrit le potentiel d'équilibre électrochimique d'une pile. Cependant, à cause de pertes irréversibles, la tension réelle aux bornes d'une pile sera toujours plus basse que celle calculée par l'équation de Nernst. Du, Li et Gu (2007) rapportent une équation d'Appleby et Foulkes (équation (1.5)) qui démontre ces pertes irréversibles.

$$E = E^\circ + \frac{RT}{F} \ln \left(\frac{a_A a_{H^+}}{a_D} \right) = E^\circ + \frac{RT}{F} \ln \left(\frac{a_A}{a_D} \right) - \frac{2.3RT}{F} pH \quad (1.4)$$

Tirée de Vidal-Iglesias et al. (2013)

$$V_{cell} = E_{cathode} - |\eta_{act,c} + \eta_{conc,c}| - E_{anode} - |\eta_{act,a} + \eta_{conc,a}| - iR_{int} \quad (1.5)$$

Tirée de Du, Li et Gu (2007)

Dans l'équation (1.5), η_{act} représente les pertes d'activation à la cathode (indice c) et à l'anode (indice a), et η_{conc} représente les pertes de concentration d'une PCM à la cathode (indice c) et à l'anode (indice a). Le terme iR_{int} indique les pertes ohmiques de la pile, où R_{int} représente la résistance interne d'une PCM, et i représente le courant électrique qui y passe.

Ainsi, il est possible d'augmenter la tension de sortie d'une pile en minimisant ces pertes. On pourra y arriver au niveau de la conception de la pile. Le choix des matériaux, du système d'échange de protons, ou de la topologie utilisée, l'ajout d'une solution tampon pour contrôler le pH, ainsi que bien d'autres paramètres, constituent tous des facteurs pouvant améliorer l'efficacité et augmenter la tension de sortie d'une PCM (Du, Li et Gu, 2007). En outre, les conditions d'opérations des PCM devront être contrôlées. Le choix du substrat à l'anode, son débit, et sa concentration viendront influencer la puissance de sortie, de même que les températures des chambres cathodique et anodique.

Cependant, lorsque les PCM disponibles sont déjà conçues et ne sont plus modifiables, l'optimisation de la puissance utilisable se fera au niveau de la méthode de récolte de l'énergie. Comme il a été expliqué à la section 1.2.1, les électrons générés par une PCM voyagent au travers de la charge connectée à ses bornes. Dans un contexte réaliste, une PCM alimenterait un circuit électronique par l'entremise d'un convertisseur CC/CC. Or, ce circuit est vu par la PCM comme étant une charge avec une certaine impédance. De façon simpliste, on pourrait modéliser ce type de système comme étant une source de tension (OCV) avec une résistance interne (R_{int}) connectée à une charge externe (R_{ext}) (Logan, 2008), où OCV représente la tension en circuit ouvert de la PCM (« Open Circuit Voltage » en anglais). Ainsi, on peut calculer la puissance de ce modèle en fonction de sa charge selon l'équation (1.6). La Figure 1.4 donne un exemple de la courbe de puissance de ce type de système lorsque la charge varie.

$$P_{out} = \frac{OCV^2 * R_{ext}}{(R_{int} + R_{ext})^2} \quad (1.6)$$

Tirée de Logan (2008)

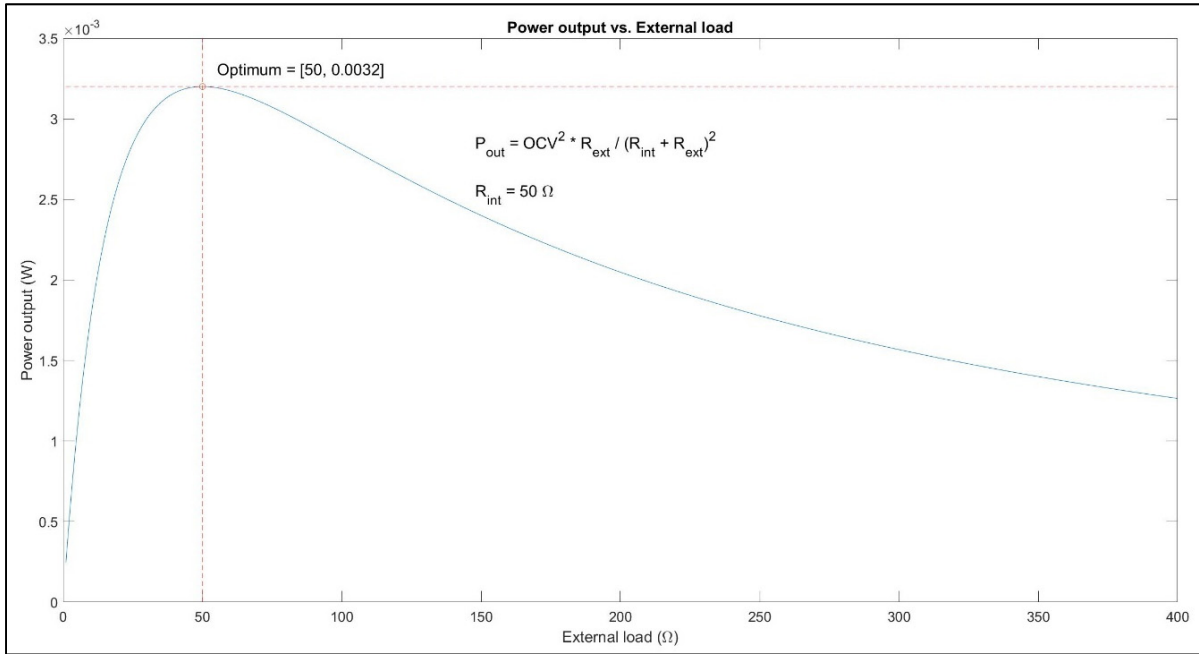


Figure 1.4 Puissance de sortie d'une source de tension avec une résistance interne en fonction de sa charge externe

Dans cet exemple, la résistance interne de la source de tension est de 50 Ω et son OCV est de 0.8 V. On peut voir que le point maximal de puissance de ce système survient lorsque $R_{ext} = R_{int}$. De la même façon, une impédance qui n'est pas adaptée à la résistance interne d'une PCM peut causer des pertes de puissance en sortie de plus de 50% (Woodward, 2009), (Wang, Park et Ren, 2012). En outre, comme dans toute source d'énergie renouvelable, des perturbations environnementales pourraient survenir et changer le point d'opération optimal d'une PCM. Par exemple, la concentration de substrat pourrait varier, ce qui occasionnerait un changement de la résistance interne d'une pile. Il faudra donc une méthode qui puisse adapter l'impédance d'un circuit externe en temps réel pour qu'elle soit toujours égale à la résistance interne de la PCM à laquelle le circuit est branché.

Afin de résoudre ce problème, il faut se tourner vers le domaine de l'optimisation en temps réel. En effet, un algorithme d'optimisation peut être utilisé afin de trouver et de suivre le point de puissance maximal d'un système. D'ailleurs, Molognoni et al. (2014) ont appliqué l'optimisation par perturbations et observations sur une PCM afin de diminuer le temps de stabilisation initial (Molognoni et al., 2014). Ils ont comparé l'utilisation d'une méthode de suivi du point de puissance maximal (« Maximum power point tracking » en anglais, ou MPPT) à l'utilisation d'une résistance fixe et leurs résultats ont démontré que l'utilisation d'une méthode MPPT a diminué d'un mois le temps nécessaire pour démarrer une MFC, et ils ont aussi noté une augmentation d'efficacité coulombique de 40%.

L'utilisation d'une méthode MPPT est aussi faite dans le cadre de ce mémoire, à l'exception que le but en est l'optimisation en temps réel de la puissance lorsque les PCM ont déjà dépassé leur stade de stabilisation initial. Dans un contexte réel, la méthode MPPT serait utilisée pour adapter l'impédance d'un circuit de conversion à la résistance interne d'une PCM en ajustant en temps réel le cycle de fonctionnement (« duty cycle », en anglais) du convertisseur. L'efficacité du convertisseur CC/CC est un facteur critique dans l'obtention de la puissance maximale de la PCM. Or, ce mémoire se concentre sur la méthode de suivi du point de puissance maximal d'une PCM. Ainsi, l'étape de conversion électrique n'est pas considérée et une résistance variable est connectée directement aux bornes des PCM optimisées, de manière à n'avoir qu'une variable à contrôler. La section 1.3 présente deux techniques d'optimisation actuellement utilisées dans le domaine des ÉR et oriente les explications sur le choix de la méthode proposée dans ce mémoire.

1.3 Optimisation en temps réel

Les énergies renouvelables sont sujettes à des contraintes dynamiques, car les systèmes utilisés pour récolter l'énergie se retrouvent dans des milieux propices à des changements environnementaux. Il sera alors question d'utiliser des méthodes d'optimisation en temps réel, vu qu'il est difficile de prévoir à quel instant surviendront ces perturbations. En outre, dans un problème statique, le temps d'attente d'une itération à l'autre ne dépendra que de la

vitesse du système utilisé (processeur, mémoire vive, programmes utilisés, etc.), tandis que dans un système dynamique, il faut attendre que la dynamique du système asservi se soit stabilisée après chaque itération afin de résoudre le problème d'optimisation statique. Le choix de l'algorithme repose donc, entre autres, sur trois critères : 1) sa vitesse de convergence, 2) son niveau de précision à trouver l'optimum d'une fonction objectif, et 3) sa capacité d'adaptation lorsque des perturbations externes surviennent.

La présente section explique d'abord l'optimisation en général et comment elle peut être adaptée pour optimiser des systèmes dynamiques en temps réel. Deux méthodes grandement utilisées comme MPPT dans le domaine des ÉR sont présentées, soient : l'optimisation par perturbations et observations (P&O), et l'optimisation par essaims particulaires (OEP). L'optimisation par P&O a été choisie dans cette étude comme base de comparaison des performances des différents algorithmes. En effet, cette méthode est largement utilisée dans le domaine des ÉR, offre de bonnes performances, et a déjà été appliquée sur des PCM. L'OEP, quant à elle, est aussi largement utilisée dans le domaine des ÉR, et sa supériorité par rapport à l'optimisation par P&O a plusieurs fois été démontrée dans des articles scientifiques. Cependant, l'OEP n'a pas encore été appliquée sur des PCM. Ainsi, l'étude vise à étudier les performances de l'OEP lorsqu'appliquée sur des PCM et à comparer ses performances à celles de l'optimisation par P&O.

1.3.1 Optimisation

L'optimisation en elle-même est un secteur des mathématiques grandement étudié dans plusieurs domaines. En effet, le but de l'optimisation est de trouver la meilleure solution à une fonction objectif. Un exemple très connu d'optimisation est le problème du voyageur de commerce (« traveling salesman problem » en anglais). Ce problème est énoncé de la façon suivante : en ayant une liste de villes à visiter et les distances qui séparent chacune d'entre elles, trouver le trajet le plus court qu'un vendeur empruntera dans le but de visiter chaque ville de la liste une seule fois. La Figure 1.5 donne un exemple d'une solution à ce problème. *A priori*, pour une liste donnée de villes à visiter, ce problème n'aura généralement qu'une

seule solution optimale car les conditions données sont statiques. Ainsi, la méthode la plus simple à utiliser pour résoudre le problème serait une recherche exhaustive, où chaque possibilité est analysée et la meilleure option est choisie. Cependant, si la liste de villes à visiter augmente de taille, l'espace de recherche du problème augmente exponentiellement et les ressources nécessaires pour le résoudre augmentent tout autant, rendant ainsi la recherche exhaustive difficile ou même impossible. Si on ajoute en plus des contraintes dynamiques au problème (e.g. : possibles embouteillages sur les routes, limites de vitesses différentes, routes fermées, etc.), la complexité du problème augmente grandement et des techniques d'optimisation plus élaborées sont alors requises.

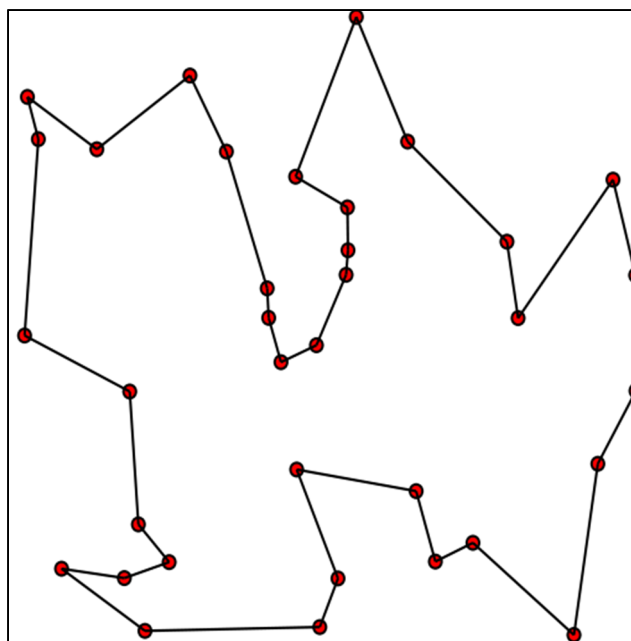


Figure 1.5 Exemple d'une solution au problème du voyageur de commerce

Plusieurs méthodes d'optimisation ont été développées au fil des années pour résoudre des problèmes de complexités diverses. Parmi ces méthodes, plusieurs d'entre elles se basent sur un modèle afin d'évaluer mathématiquement la solution optimale et d'y amener le système. Lorsque ces méthodes d'optimisation sont appliquées sur un système dynamique, on parlera alors d'optimisation en temps réel (OTR). Cependant, de nouveaux défis apparaissent lorsque

l'optimisation est faite dans un milieu dynamique. En effet, les paramètres calculés d'un modèle mathématique peuvent varier dans le temps, rendant ainsi le modèle initial erroné. Afin d'y remédier, certaines méthodes adaptatives ajustent les paramètres de leur modèle de façon concourante à l'optimisation en utilisant les données recueillies afin d'obtenir un modèle plus réaliste, et ce en temps réel. Cependant, afin de résoudre les problèmes où l'obtention d'un modèle mathématique serait trop complexe, voire infaisable, il est plus avantageux d'utiliser des techniques d'optimisation ne requérant aucune connaissance préalable du modèle du système asservi. Ces méthodes considéreront le système à optimiser comme étant une boîte noire et l'optimisation sera faite de façon itérative en faisant une recherche partielle de l'hyperespace de la fonction objectif et une solution sera trouvée. Ainsi, si une perturbation survient, il ne faudra se fier qu'à la capacité d'adaptation de l'algorithme pour trouver la nouvelle position de l'optimum dans l'hyperespace, sans devoir se soucier de réajuster les paramètres d'un modèle rendu obsolète par la perturbation dynamique.

Dans le domaine des ÉR, les techniques d'OTR utilisées pour trouver et suivre le point de puissance maximal d'un système sont appelées des techniques de suivi du point de puissance maximal (« Maximum power point tracking » en anglais, ou MPPT). Une grande famille des techniques d'MPPT sont les méthodes basées sur l'approche de la plus forte pente avec une estimation du gradient. On retrouve plusieurs algorithmes dans ce type d'optimisation. La méthode de perturbations et observations (P&O) est grandement utilisée dans le domaine des ÉR car elle est très simple et réussit à trouver l'optimum d'une fonction objectif de façon fiable. Cette méthode a souvent été utilisée pour optimiser des systèmes PV (Sridhar et al., 2010), (Azzouzi, 2012), (Khaehintung, Wiangtong et Sirisuk, 2006), et elle a aussi été appliquée sur des PCM (Woodward et al., 2010), (Pinto et al., 2011). La commande extrême (« Extremum seeking control » en anglais, ou ESC) est aussi grandement prisee pour sa capacité d'adaptation aux changements de gradient dans une fonction objectif (Leyva et al., 2006), (Li et al., 2011). L'optimisation par multi-unités (OMU) (Srinivasan, 2007) s'apparente à l'ESC, mais utilise plusieurs unités ayant une courbe statique de puissance semblable ou identique afin d'amener le système à converger plus rapidement vers

l'optimum d'une fonction objectif. L'OMU a été appliquée sur des PCM (Woodward et al., 2009) et des simulations sur un parc éolien ont donné de bons résultats (Mehenna et Woodward, 2012), (Guillemette et Woodward, 2014).

Afin de diminuer le temps requis pour trouver l'optimum d'une fonction objectif, il peut être avantageux de faire une recherche aléatoire de l'espace de recherche afin de trouver rapidement la région dans laquelle se trouve l'optimum. Dans cette optique, les algorithmes métaheuristiques peuvent être intéressants. L'une des caractéristiques principales des métaheuristiques sont leur nature souvent stochastique. En effet, en introduisant un aspect aléatoire à la recherche de l'hyperespace, il est possible d'explorer une plus grande région de l'espace de recherche et de discréditer les régions moins intéressantes plus rapidement pour concentrer les recherches là où la fonction objectif est meilleure. L'optimisation par essaims particuliers (OEP) s'inscrit dans ce type de recherche stochastique. Grandement appréciée pour sa simplicité d'implémentation et sa rapidité de convergence, l'OEP est largement utilisée comme technique d'MPPT dans les systèmes PV (Sundareswaran, Vignesh kumar et Palani, 2015), (Huynh et al., 2013), (Ishaque et al., 2012), (Chassé et Woodward, 2016).

Avec autant de possibilités comme choix de méthode d'MPPT, il est impératif d'évaluer les avantages et inconvénients de chacune d'entre elles afin de décider laquelle utiliser pour un système à optimiser. La section 1.3.2 se concentre sur le P&O, puis la section 1.3.3 élabore plus en détails en quoi consiste l'OEP.

1.3.2 Optimisation par perturbations et observations

La méthode P&O classique opère un système dynamique en appliquant périodiquement une perturbation (Δ) à l'entrée de ce système (Liu et Lopes, 2004). Le résultat en sortie est ensuite observé et, si la valeur de la fonction objectif obtenue suite à cette perturbation est supérieure (dans le cas d'une maximisation) à celle obtenue à la perturbation précédente, une nouvelle perturbation de même signe que la précédente est appliquée. Si, au contraire, la valeur obtenue est inférieure à celle de l'itération précédente, une perturbation de signe

opposé est appliquée $(-\Delta)$. L'équation (1.7) donne le calcul de la prochaine position d'une unité optimisée par P&O.

$$d(i + 1) = d(i) + k * \Delta \quad (1.7)$$

avec d la position de l'unité à l'itération i , k le sens de la perturbation appliquée sur l'unité, et Δ l'amplitude de la perturbation. Le système convergera ainsi vers l'optimum de la fonction objectif et oscillera autour de ce point. Cette méthode est très pratique car elle ne nécessite pas la connaissance préalable d'un modèle fondamental pour le système que l'on veut asservir. Cependant, la vitesse de convergence ainsi que le niveau d'oscillations en régime permanent dépendent directement du paramètre Δ . En effet, un plus grand Δ augmentera la vitesse de convergence, mais augmentera aussi l'amplitude des oscillations autour de l'optimum, réduisant ainsi l'efficacité nette de l'algorithme. La Figure 1.6 montre un exemple d'application du P&O sur une fonction objectif définie par l'équation (1.8) avec $\Delta = 10$. On observe que l'algorithme prend à peu près 20 itérations pour converger, ce qui est relativement long, et que les oscillations autour de l'optimum sont de faible amplitude. Il est à noter que le temps de convergence aurait été encore plus long si la condition initiale était plus loin de l'optimum. Ainsi, en appliquant le P&O sur un système dynamique, l'amplitude des perturbations dicterait la vitesse de convergence. Cependant, dans un environnement où les perturbations externes n'influencent pas beaucoup la position de l'optimum, le P&O est une stratégie d'optimisation valide.

$$J = -5.5 * 10^{-5} * (d - 325)^2 + 31 \quad (1.8)$$

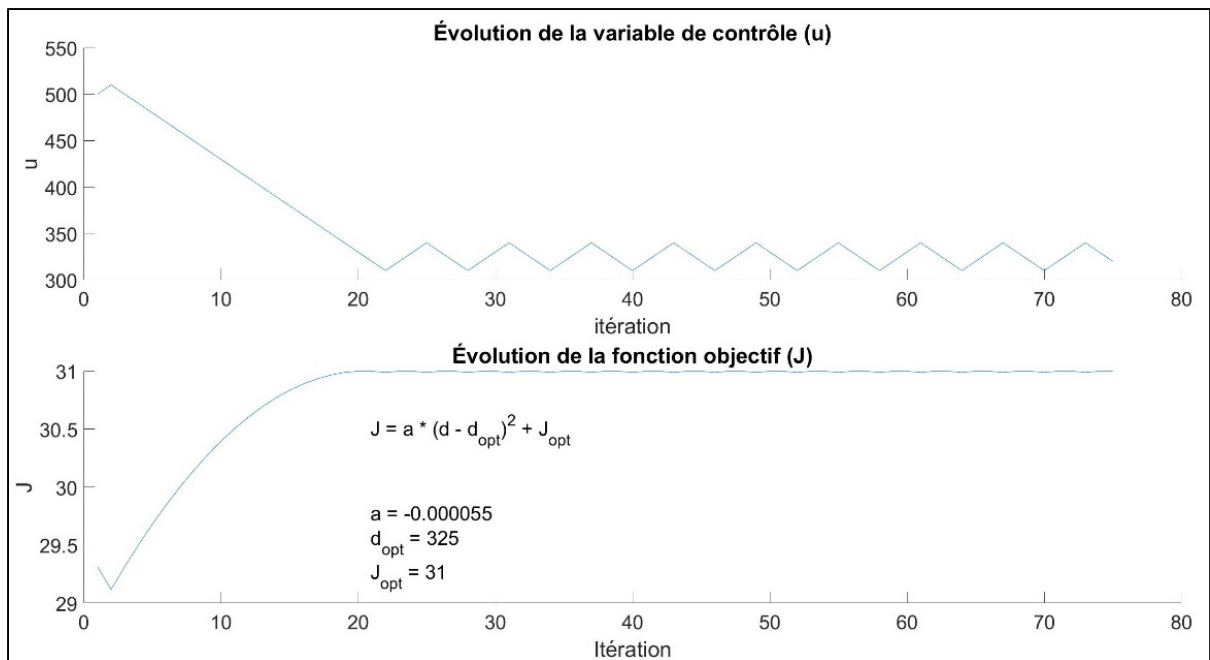


Figure 1.6 Application du P&O sur une fonction objectif statique

1.3.3 Optimisation par essais particuliers

L'optimisation par essais particuliers (OEP) a été développée en 1995 (Eberhart et Kennedy, 1995). Cet algorithme est basé sur le comportement d'une volée d'oiseaux. Dans une volée d'oiseaux, chaque individu choisit sa prochaine position en étant influencé par le reste de l'essaim. La direction générale de la volée, l'inertie de l'individu, ainsi que sa position spatiale viendront influencer son choix dans ses prochains mouvements. L'OEP applique ce principe dans le but de trouver l'optimum d'une fonction objectif. Pour ce faire, l'OEP implémente une population d'individus qui navigueront dans un espace de recherche afin de trouver une position optimale tout en tenant compte des informations récoltées par leurs voisins. L'OEP est intéressante dans ce sens car les informations récoltées lors de la recherche d'extremum sont partagées dans toute la population, aidant ainsi l'algorithme à converger plus rapidement.

Ces individus, dits « particules », sont initialisés à une position aléatoire dans l'espace de recherche, ou hyperspace, avec une vitesse initiale aléatoire. Cette vitesse déterminera la

prochaine position d'une particule. Les vitesses subséquentes sont calculées selon trois opérandes : l'opérande inertiel de la particule, son opérande cognitif, et son opérande social. L'équation (1.9) présente l'équation de la vitesse d'une particule, et l'équation (1.10) montre le calcul de la prochaine position d'une particule.

$$\begin{aligned} v(i) = & [\omega * v(i - 1)] \\ & + [c_1 * r_1(i) * (Pbest(i) - d(i))] \\ & + [c_2 * r_2(i) * (Gbest(i) - d(i))] \end{aligned} \quad (1.9)$$

$$d(i + 1) = d(i) + v(i) \quad (1.10)$$

L'opérande inertiel est calculé avec le coefficient inertiel de la particule ω et sa vitesse précédente $v(i - 1)$, soit $\omega * v(i - 1)$. À noter que ω doit être inférieur à un et supérieur ou égal à zéro. Dans le cas contraire, cela pourrait amener le système à ne jamais converger et à toujours rester en oscillation. L'opérande cognitif est calculé en fonction de la différence entre la meilleure position de la particule dans les itérations précédentes ($Pbest(i)$, pour « Personal best ») et sa position courante $d(i)$, son coefficient cognitif c_1 , et un nombre aléatoire $r_1 \in [0,1]$. L'opérande social est calculé en fonction de la différence entre la meilleure position de toutes les particules de l'essaim ($Gbest(i)$, pour « Global best ») et sa position courante d_i , son coefficient social c_2 , et un nombre aléatoire $r_2 \in [0,1]$. Ainsi, l'OEP pourrait être utilisée comme méthode d'MMPT pour les PCM car toutes les particules du système convergeront vers l'optimum d'une fonction objectif.

Il est à noter que, bien qu'un essaim particulaire de l'OEP comprenne plusieurs particules, cet algorithme est utilisé pour optimiser la fonction objectif d'un seul système, que l'on peut considérer comme étant une unité. Or, une itération d'OEP nécessite une mesure de la fonction objectif de toutes les particules du système. Ainsi, afin de créer une population de particules, il faudra analyser une particule à la fois, que l'on dénotera « particule active ». Pour ce faire, on amène le système à la position de la particule active et, lorsque le système

est stable (dans le cas d'un système dynamique), on mesure la valeur de la fonction objectif, que l'on associe à la particule active, puis on passe à la particule suivante.

La Figure 1.7 présente l'ordinogramme de l'OEP. À chaque itération, on positionne séquentiellement la variable de contrôle du système à la position de la particule active et on évalue la fonction objectif afin d'obtenir une valeur de fitness pour la particule. On compare ensuite le Pbest de chaque particule à leur valeur de fitness courante afin d'évaluer s'il faut mettre à jour Pbest. Lorsque toutes les particules ont été évaluées, le Gbest de l'essaim est ensuite évalué, puis les prochaines vitesses des particules sont calculées avec l'équation (1.9), et les prochaines positions sont calculées avec l'équation (1.10). Le cycle est ensuite répété jusqu'à ce qu'une condition de fin soit atteinte, e.g. un nombre d'itérations.

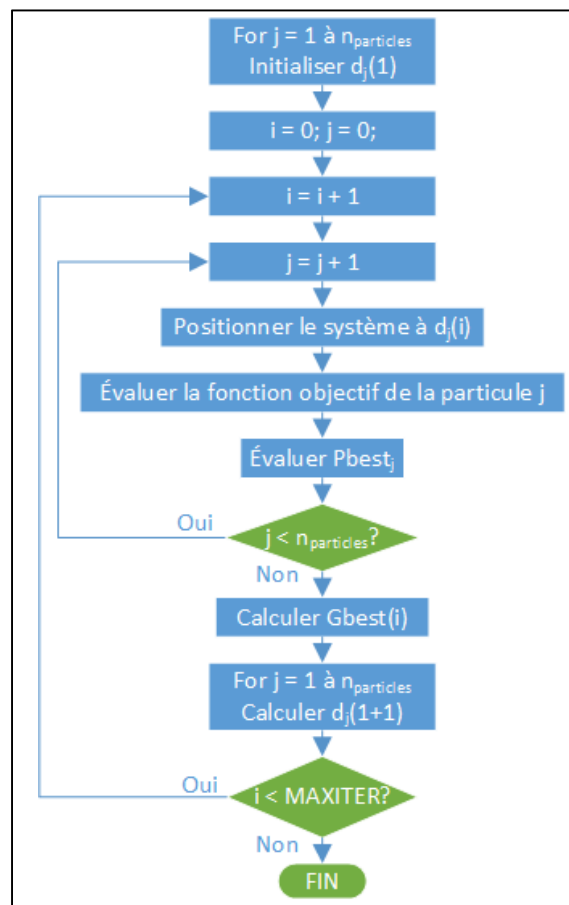


Figure 1.7 Ordinogramme de l'optimisation par essais particulaires

Afin d'illustrer graphiquement l'évolution d'une optimisation faite par OEP, la Figure 1.8 démontre l'application de l'OEP sur une fonction objectif définie par l'équation (1.8). Dans cet exemple, l'essaim compte trois particules, et les coefficients sont : $\omega = 0.4$, $c_1 = 1$, et $c_2 = 2$. Dans cet exemple, on note que l'algorithme converge en à peu près 30 itérations, ce qui est plutôt lent. Cependant, il n'y a presque pas d'oscillations autour de l'optimum, ce qui augmente l'efficacité de l'algorithme.

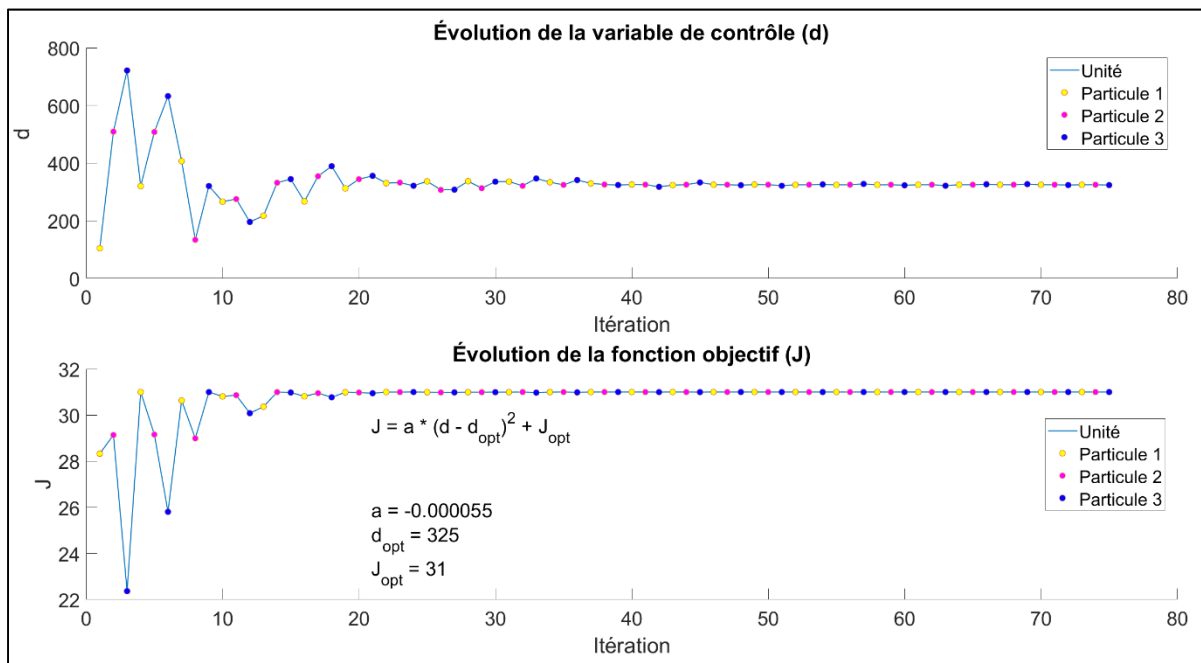


Figure 1.8 Simulation de l'OEP sur une fonction objectif statique

L'OEP est peu coûteuse en termes de mémoire et de vitesse en raison de la simplicité de son algorithme (Eberhart et Kennedy, 1995). Ainsi, elle peut être facilement implémentée dans un microcontrôleur, ce qui la rend avantageuse dans le cas où l'on voudrait l'utiliser dans un système embarqué. D'ailleurs, Ishaque et al. (2012) ont utilisé un DSP de Texas Instruments pour asservir une matrice de cellules photovoltaïques avec une technique d'OEP modifiée (Ishaque et al., 2012). En outre, l'OEP présente peu d'oscillations en régime permanent, car les vitesses des particules, une fois l'extremum trouvé, sont presque nulles.

L'OEP comporte néanmoins quelques inconvénients. Notamment, si la fonction objectif présente plusieurs extrema locaux, elle est à risque de converger vers un extremum local. Aussi, le système ne réagit pas bien lorsque des perturbations surviennent dans l'environnement. Ainsi, pour corriger la situation, plusieurs adaptations à cette méthode ont été faites depuis quelques années. Par exemple, l'ajout de particules sentinelles a été fait dans le but de détecter les perturbations dans l'environnement (Nguyen, Yang et Branke, 2012). Lorsqu'une sentinelle détecte une perturbation, l'essaim est réinitialisé et une nouvelle recherche commence. En 2004, l'OEP dynamique a été proposée, où les coefficients c_1 et c_2 varient dans le temps afin de mettre plus de poids à $Pbest$ au début de l'optimisation, pour ensuite mettre plus de poids à $Gbest$ à la fin de l'optimisation (Ratnaweera, Halgamuge et Watson, 2004). Cette modification diminue le risque que les particules convergent vers un optimum local au lieu de l'optimum global. L'OEP dynamique a été appliquée sur un système PV (Huynh et al., 2013) et les résultats ont démontré une efficacité de suivi du point maximal de puissance accrue par rapport à l'OEP classique. En 2015, Chen et al. ont proposé une approche parallèle à l'OEP avec l'ajout d'un classificateur pour optimiser des panneaux PV (Chen et al., 2015). En 2016, nous avons comparé les performances de l'OEP parallélisée à celles de l'OEP classique dans des systèmes PV, et il a été prouvé qu'il peut être avantageux d'optimiser chaque unité d'un système individuellement plutôt que d'optimiser le même système comme un tout (Chassé et Woodward, 2016). Les résultats ont aussi démontré que l'OEP parallèle convergeait beaucoup plus rapidement que l'OEP classique. Nous avons néanmoins noté que l'efficacité de l'algorithme chute grandement dans le cas où les unités présentent des courbes de puissance largement différentes.

Les algorithmes étudiés dans ce mémoire sont basés sur l'OEP, avec certaines modifications permettant d'adapter les algorithmes à un contexte dynamique, et leurs performances sont comparées à celles du P&O. Le CHAPITRE 2 élabore davantage sur les algorithmes étudiés, puis le CHAPITRE 3 et le CHAPITRE 4 valident les algorithmes sur des PCM et sur des cellules photovoltaïques, respectivement.

1.4 Conclusion

Les PCM utilisent des microbes pour faire l'oxydation de matières organiques. Dans le procédé, de l'eau propre est créée et de l'électricité est générée. Dans cette optique, les PCM seraient utiles, entre autres, dans le domaine du traitement des eaux usées. Cependant, comme elles ne fournissent pas beaucoup de puissance, elles n'ont pas encore été intégrées dans l'industrie en tant que source d'ÉR viable. Ainsi, des recherches se font afin d'augmenter l'énergie fournie par ces piles. Afin d'y parvenir, les recherches vont généralement s'orienter autour de deux aspects, soient : la conception de la pile, ou les modes d'opération des PCM. C'est dans l'optique du mode d'opération d'une PCM que ce mémoire s'oriente, dans le sens où plusieurs méthodes d'optimisation en temps réel sont évaluées dans le but d'optimiser la puissance de sortie de piles à combustible microbiennes.

L'OEP est un algorithme largement utilisé dans l'optimisation de système photovoltaïques, tout comme le P&O. Cependant, l'OEP n'a pas encore été appliquée sur un système composé de PCM, contrairement au P&O. Ainsi, ce mémoire tente de valider la capacité de l'OEP à optimiser un tel système en comparant les performances de l'algorithme à celles du P&O. En outre, un algorithme d'optimisation en temps réel basé sur l'OEP est proposé, et ses performances sont comparées à celles de l'OEP et du P&O. Le CHAPITRE 3 présente des résultats de simulations et des résultats expérimentaux sur des PCM, et le CHAPITRE 4 compare les performances de l'algorithme proposé lorsqu'appliqué sur un système PV à celles de l'OEP et du P&O.

CHAPITRE 2

ADAPTATION DE L'OPTIMISATION PAR ESSAIS PARTICULAIRES AU PROBLÈME D'OPTIMISATION EN TEMPS RÉEL

2.1 Introduction

Comme mentionné à la section 1.3.3, l'OEP est déjà grandement utilisée comme technique de suivi du point maximal de puissance pour plusieurs systèmes, notamment pour les systèmes PV. L'OEP est intéressante en tant que méthode d'MPPT car elle converge assez rapidement et trouve précisément l'optimum d'une fonction objectif convexe. Grâce aux développements des dernières années, l'OEP s'adapte bien lorsque des perturbations environnementales surviennent (Nguyen, Yang et Branke, 2012). De plus, l'OEP ne requiert aucun modèle préalable car le système à optimiser est considéré comme une boîte noire. Cependant, selon nos recherches, cette technique n'a pas encore été appliquée aux PCM. Ainsi, ce mémoire se concentre sur l'optimisation par essais particuliers afin d'étudier la possibilité d'utiliser ce type d'algorithme comme méthode d'MPPT pour les PCM.

Ce chapitre présente d'abord des modifications à l'optimisation par essais particuliers classique nécessaires pour adapter l'algorithme d'origine à des systèmes dynamiques réels. Ensuite, l'optimisation par essais particuliers parallélisée (OEPP) (« Parallel PSO » en anglais, ou PPSO) est détaillée, puis un algorithme développé dans le cadre de ce mémoire s'inspirant de l'OEPP est élaboré. Cet algorithme est intitulé optimisation par essais particuliers parallélisée avec classificateur (OEPPC) (« Parallel PSO with Classification » en anglais, ou PPSOC).

2.2 Optimisation par essais particuliers séquentielle

L'OEP classique n'est pas très apte à optimiser des systèmes dynamiques, car l'algorithme considère seulement une fonction statique. Or, les ÉR sont des systèmes dynamiques où des perturbations peuvent survenir. Ainsi, ce mémoire définit quelques modifications sur l'OEP

classique afin d'améliorer ses performances dans des conditions dynamiques. À des fins pratiques, l'algorithme d'OEP modifiée est ainsi nommé optimisation par essais particuliers séquentielle, ou OEPS. Les modifications à l'OEP énumérées dans les sections 2.2.1 à 2.2.5 sont aussi utilisées dans les deux autres algorithmes présentés dans ce chapitre.

2.2.1 Saturation et résolution

Dans un contexte de simulation, la résolution de la variable de commande d'un algorithme peut être très précise et ses limites inférieure et supérieure sont choisies arbitrairement. Cependant, dans un contexte expérimental, la variable de commande aura une résolution et des limites dictées par le matériel utilisé. Par exemple, le banc d'essai utilisé dans ce mémoire commande des potentiomètres numériques dont les limites inférieure et supérieure sont de 50Ω et de 1050Ω respectivement, et dont la résolution est d'à peu près 3.92Ω (le banc d'essai est détaillé au CHAPITRE 3). Pour prendre en compte ces limitations, le pseudocode suivant a été ajouté à l'OEPS lors du calcul des prochaines positions des particules pour ajouter une saturation :

- calculer la vitesse de la particule j avec l'équation (1.9)
- calculer la prochaine position de la particule j avec l'équation (1.10)
- **SI** $d_j(i + 1) > d_{max}$

$$d_j(i + 1) = d_{max}$$

$$v_j(i) = d_j(i + 1) - d_j(i)$$
- **SI** $d_j(i + 1) < d_{min}$

$$d_j(i + 1) = d_{min}$$

$$v_j(i) = d_j(i + 1) - d_j(i)$$

avec d_j la position de la particule j , i l'itération courante, d_{max} la position maximale d'une particule, et d_{min} la position minimale d'une particule. Le pseudocode ci-dessous a aussi été ajouté afin de prendre en compte la résolution de la variable de contrôle :

- calculer la prochaine position de la particule j
- $reste = modulo(d_j(i + 1) - d_{min}, résolution)$
- $d_j(i + 1) = d_j(i + 1) - reste$
- **SI** $reste > (résolution / 2)$

$$d_j(i + 1) = d_j(i + 1) + résolution$$

avec $modulo(x, y)$ l'opération modulo de x par rapport à y , et $résolution$ la résolution de la variable de commande.

2.2.2 Pbest

Dans les problèmes de maximisation, la variable Pbest d'une particule j se calcule généralement de la façon suivante :

$$\begin{aligned} \text{SI } F_j(i) > F_{pbest_j} \\ F_{pbest_j} &= F_j(i) \\ Pbest_j &= d_j(i) \end{aligned}$$

avec $F_j(i)$ la valeur de fitness courante de la particule j à l'itération i , et F_{pbest_j} la valeur de fitness associée à la position $d_j = Pbest_j$.

Ainsi, $Pbest$ représente la position optimale d'une particule depuis le début de l'optimisation. Des simulations ont démontré que l'OEPS performait mieux lorsque $Pbest$ était comparé uniquement à la position précédente. Cette observation est avantageuse lorsque l'on veut optimiser un système dynamique, car dans le cas où une perturbation surviendrait, $Pbest$ n'est plus valide étant donné que la valeur de fitness à la position $Pbest$ a probablement changé. Donc, dans ce mémoire, la variable $Pbest$ d'une particule j est calculée de la façon suivante :

- **SI** $F_j(i) > F_{pbest_j}$
 $F_{pbest_j} = F_j(i)$
 $Pbest_j = d_j(i)$
- **SINON**
 $F_{pbest_j} = F_j(i - 1)$
 $Pbest_j = d_j(i - 1)$

avec F_{pbest_j} la valeur de fitness associée au $Pbest$ de la particule j .

2.2.3 Détection de perturbations et réaction à celles-ci

Hu et Eberhart (Hu et Eberhart, 2002) ont proposé une approche avec mémoire pour détecter les perturbations dans l'environnement. Leur approche consiste à réévaluer les solutions à chaque itération, de manière à ce que chaque particule soit une sentinelle qui peut détecter des changements dans la fonction objectif d'un système. Si une particule n'a pas bougé depuis la dernière itération, mais que la valeur de sa fonction objectif a changé, tout en tenant compte du bruit de mesure, cela indique qu'une perturbation est survenue sur le système.

Leur méthode de réaction aux perturbations consiste à introduire de la diversité lorsqu'un changement est détecté. Pour ce faire, la mémoire de l'algorithme est réinitialisée ($Pbest$, $Gbest$, vitesses des particules, etc.) et les particules sont repositionnées aléatoirement dans l'espace de recherche. Le pseudocode ci-dessous démontre la logique de la méthode de détection et de réaction aux perturbations choisie pour cette étude :

```

POUR  $i = 1 : nParticles$ 
     $p = \text{swarm.particles}(i)$ 
    SI  $p.curPos == p.prevPos$ 
        SI  $\text{abs}(p.curFit - p.prevFit)/p.prevFit > p.margin$ 
             $\text{swarm.Reset}()$ 
SORTIR DE LA BOUCLE

```

avec $nParticles$ le nombre de particules dans l'essaim, $swarm.particles(i)$ et p la particule i de l'essaim, $p.curPos$ et $p.prevPos$ les positions courante et précédente de la particule p dans l'hyperespace, respectivement, $p.curFit$ et $p.prevFit$ les valeurs courante et précédente de fitness de la particule p , respectivement, $p.margin$ la tolérance aux perturbations de la particule (e.g. 5%), et $swarm.Reset()$ une fonction qui réinitialise l'essaim, c'est-à-dire $Gbest$, tous les $Pbest$, ainsi que les positions et les vitesses des particules.

2.2.4 Positions initiales des particules

Il a été déterminé par simulations qu'il est avantageux de diviser initialement l'espace de recherche en un nombre de sections équivalent au nombre de particules de l'OEPS. Par exemple, pour un espace de recherche entre 50 et 200, et une OEPS à 3 particules, les positions initiales seraient : $d_1(1) \in [50, 100]$, $d_2(1) \in]100, 150]$, et $d_3(1) \in]150, 200]$.

En effet, en subdivisant l'hyperespace, il y a peu de risques que les particules soient toutes initialisées à des positions très proches les unes des autres, réduisant ainsi le risque de converger à une position non optimale. Ainsi, le pseudocode suivant est implémenté dans le calcul des positions initiales des particules :

- $range = posMax - posMin$
- $sectionLength = range / nParticles$
- $sections = []$
- $sections(1) = posMin$
- $sections(nParticles + 1) = posMax$
- **POUR** $i = 2 : nParticles$
 - $sections(i) = sectionLength * i$
- **POUR** $i = 1 : nParticles$
 - $p = swarm.particles(i)$
 - $p.curPos = rand(sections(i), sections(i+1))$

avec $posMin$ et $posMax$ les limites inférieure et supérieure de la variable de commande, respectivement, $sections$ un vecteur initialement vide d'une longueur de $(nParticles + 1)$, et $rand(arg1, arg2)$ une fonction qui retourne un nombre aléatoire compris entre $arg1$ et $arg2$.

2.2.5 Condition de fin

L'OEP classique comprend normalement un nombre d'itérations fini. Or, étant donné que l'on veut optimiser continuellement les systèmes asservis, cette condition est ignorée pour l'OEPS et l'algorithme opère continuellement sur le système.

2.3 Optimisation par essais particuliers parallélisée

Lorsque l'OEP est utilisée pour optimiser un système photovoltaïque, elle est généralement appliquée en traitant le système comme une entité. Un nombre de particules sera défini, une position initiale sera associée à chaque particule, et la variable de commande du système sera ajustée à la position d'une particule à la fois afin d'obtenir séquentiellement une valeur de fitness pour chaque position. Lorsque toutes les particules ont une valeur de fitness qui leur est associée, de nouvelles positions pour les particules sont calculées et une nouvelle itération de l'algorithme est démarrée. Étant donné la dynamique très rapide des cellules PV, le temps de stabilisation du système d'un point d'opération à un autre est très court. Cependant, pour des systèmes plus lents, tels que les PCM, il faut attendre plusieurs minutes avant que le système ne se soit stabilisé.

Dans notre article (Chassé et Woodward, 2016), nous avons démontré qu'il est avantageux d'optimiser les différentes unités d'un système en parallèle plutôt que de les optimiser comme un tout, i.e. appliquer une MPPT sur toutes les cellules PV d'un panneau PV individuellement plutôt qu'appliquer une MPPT sur le panneau en entier. La méthode choisie dans notre article a été l'optimisation par essais particuliers parallélisée (OEPP). Cet algorithme s'applique lorsque le système à optimiser comporte plusieurs unités semblables ou identiques, e.g. plusieurs cellules solaires ou bien un réseau de plusieurs PCM. L'idée derrière cet algorithme peut être comparée au principe de l'optimisation par multi-unités

(OMU), brièvement décrite à la section 1.3.1. En effet, l'OMU prend avantage des similarités entre les courbes statiques de puissance de deux unités afin d'accélérer la convergence aux optima. Dans la même optique, l'OEPP sera appliquée sur un système comprenant n unités, et chaque unité sera considérée comme une particule. En procédant de la sorte, la vitesse de convergence de l'OEPP peut être plus rapide que celle de l'OEP d'un facteur n , n étant le nombre d'unités du système ainsi que le nombre de particules de l'algorithme.

L'idée de considérer chaque unité d'un système comme une particule a été premièrement étudiée par Chen et al. (Chen et al., 2015). Dans leur étude, Chen et al. ont appliqué un algorithme d'OEP modifié sur plusieurs panneaux PV en considérant chaque panneau comme une particule, pour ensuite regrouper les panneaux présentant des caractéristiques semblables dans différents essaims particulaires. Les résultats ont été comparés à ceux du P&O sur le même système, et leur application de l'OEP a démontré un temps de convergence réduit ainsi qu'une plus faible oscillation en régime permanent. Dans notre article, nous avons comparé les performances de l'OEPP appliquée sur chaque cellule d'un panneau PV à celles de l'OEPS appliquée sur le même panneau solaire lorsque toutes les cellules étaient reliées en série (Chassé et Woodward, 2016). Nous avons dénoté un temps de convergence pour l'OEPP moins long d'un facteur n , où n est le nombre de cellules dans le panneau PV, et donc aussi le nombre de particules d'un essaim d'OEPP. Il a aussi été observé que la puissance totale finale était plus élevée pour l'OEPP que pour l'OEPS.

La Figure 2.1 montre les ordinogrammes de l'OEPS et de l'OEPP, avec $n_{particles}$ le nombre de particules d'un essaim, j l'index d'une particule, i l'index de l'itération, $d_j(i)$ la position dans l'hyperespace de la particule j à l'itération i , et $MAXITER$ le nombre d'itérations total de l'algorithme, défini par l'utilisateur. À noter que $MAXITER$ n'est présenté ici que pour dicter un nombre d'itérations fini pour les simulations. Comme il a été mentionné à la section 2.2.5, les algorithmes opèrent normalement continuellement, sans condition de fin. À des fins de comparaison, la variable k a été ajoutée afin de compter le nombre de fois où l'algorithme a dû attendre que le système se stabilise avant de pouvoir changer de point d'opération. Il faut se rappeler que, dans un système dynamique, il faut attendre que le système se soit

stabilisé après chaque itération afin d'obtenir une bonne représentation de la courbe statique de puissance des unités afin d'ignorer le régime transitoire. Ainsi, la vitesse de convergence de l'algorithme est l'un des critères les plus recherchés, car, pour les MPPT, une convergence plus rapide se traduit en une plus grande quantité d'énergie récoltée.

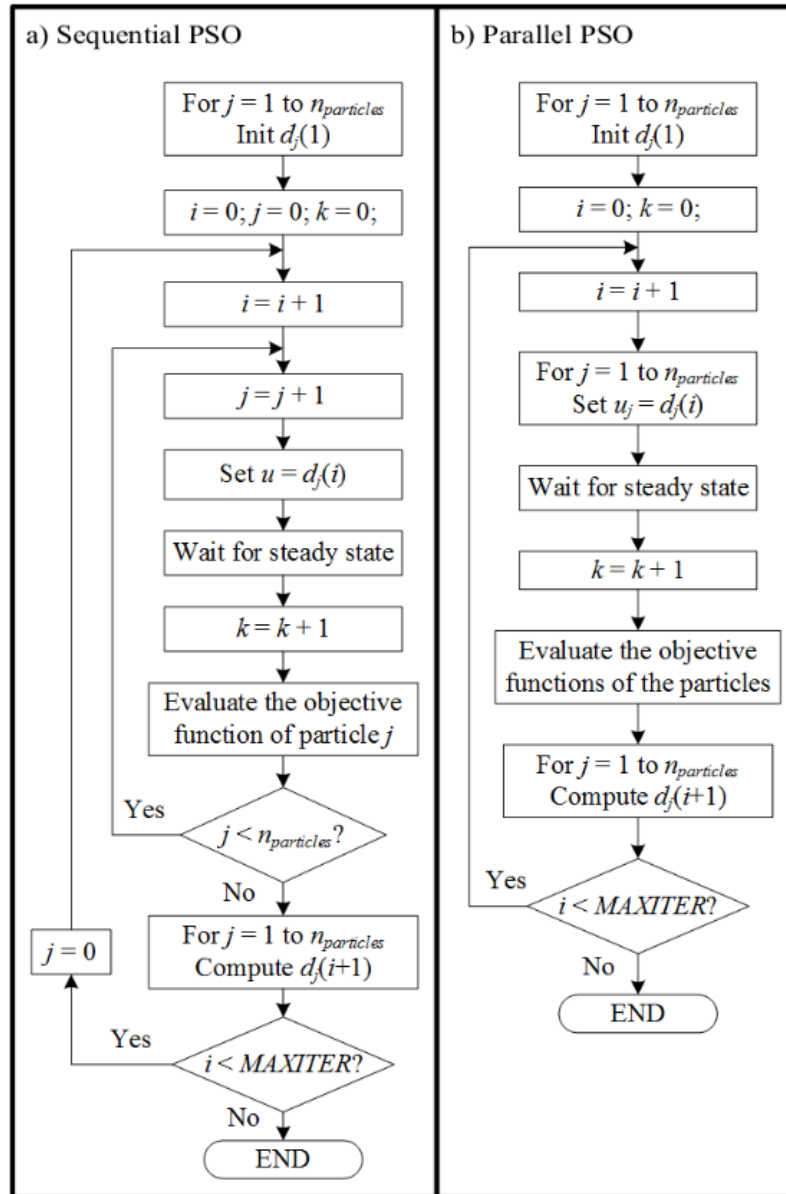


Figure 2.1 Ordigrammes de a) l'OEP séquentielle,
et de b) l'OEP Parallélisée
Tirée de Chassé et Woodward (2016)

Étant donné que les valeurs de fitness des particules sont évaluées en parallèle, la variable k est n fois moins grande pour l'OEPP que pour l'OEPS à la fin de l'optimisation, où n est le nombre de particules. Ainsi, pour un même nombre d'itérations total, les unités de l'OEPP ont été à leur optimum plus longtemps que les unités de l'OEPS. Comme on peut le voir à la Figure 2.1, les particules sont toutes évaluées en parallèle pour l'OEPP tandis qu'elles sont évaluées de façon séquentielle pour l'OEPS. C'est de cette façon que l'on peut diminuer le temps de convergence de l'algorithme d'un facteur n .

Il est vrai que pour certains systèmes, le temps de stabilisation est très court, donc l'avantage de vitesse de l'OEPP n'est pas vraiment perçu. Par exemple, les cellules PV se stabilisent en quelques microsecondes. Cependant, dans le cas des PCM, il faut attendre plusieurs minutes lorsque l'on change le point d'opération. Ainsi, l'avantage de l'OEPP est vraiment mis de l'avant pour les PCM.

Néanmoins, nous avons constaté que, dans le cas où les unités d'un système présentent des courbes de puissance statiques trop différentes, l'efficacité de l'algorithme pourrait grandement diminuer car certaines particules convergeraient vers un point qui n'est possiblement pas leur optimum. La Figure 2.2, tirée de (Woodward, 2009), montre un exemple de deux fonctions objectif non identiques. Woodward note les différences entre les optima des unités comme β la différence des positions optimales des unités, et γ la différence des valeurs de fitness optimales. Nos tests sur l'OEPP ont démontré qu'une différence en γ n'influence pas beaucoup la vitesse ni la précision de l'algorithme, tandis qu'une différence en β affecte la capacité de l'OEPP à trouver la position optimale de chaque unité. En effet, le calcul de la prochaine position d'une particule (équations (1.9) et (1.10)) est influencé par $Gbest$ et $Pbest$, qui représentent des positions optimales. Donc, dans le cas où $\beta = 0$ et $\gamma \neq 0$, les particules convergeront tout de même à leur optimum respectif étant donné que la position de tous les optima dans l'hyperespace est la même. Cependant, dans le cas où $\beta \neq 0$, les particules convergeront vers la position optimale de la particule ayant la meilleure valeur de fitness. Comme il est souvent non pratique de connaître préalablement les courbes de puissance des unités asservies, l'OEPP n'est pas idéale comme MPPT pour optimiser des

unités pouvant être non identiques. Ainsi, l'algorithme proposé à la section 2.4 tente de compenser cet inconvénient tout en gardant l'avantage de vitesse que présente l'OEPP.

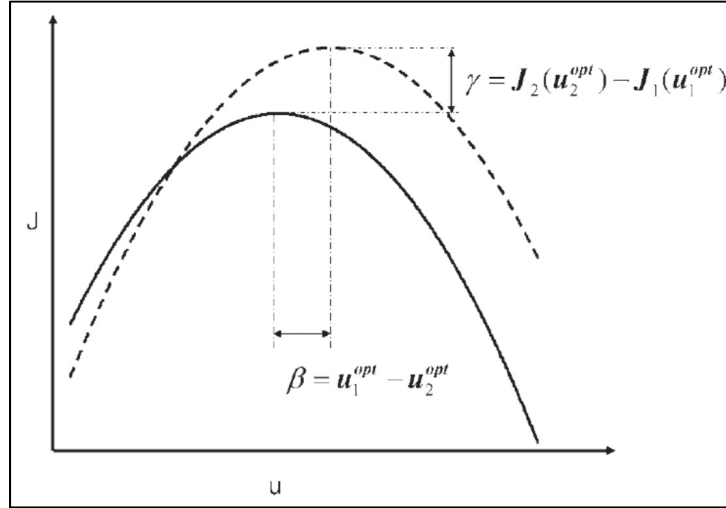


Figure 2.2 Différences statiques entre deux unités
Tirée de Woodward (2009)

2.4 Optimisation par essais particuliers parallélisée avec classificateur

L'algorithme d'optimisation par essais particuliers avec classificateur (OEPPC) a été développé dans le cadre de ce mémoire dans l'optique de prendre avantage de la vitesse de convergence très rapide de l'OEPP pour optimiser des systèmes dynamiques. La première étape de l'OEPPC consiste à appliquer l'OEPP sur toutes les unités afin d'obtenir une convergence rapide. Cependant, comme il a été mentionné à la section 2.3, les performances de l'OEPP diminuent grandement lorsque les unités à optimiser présentent des grandes différences au niveau de leur courbe de puissance statique. Cette lacune est comblée par l'OEPPC grâce à certaines modifications à l'algorithme. Ces ajouts sont présentés ci-bas, puis plus en détail dans les sous-sections 2.4.1.1 à 2.4.1.3.

Intégration de l'algorithme P&O

Tout d'abord l'ajout du P&O à l'algorithme permet de faire un ajustement fin des positions finales des unités après la convergence de l'OEPP. En effet, dans le cas où une unité, après avoir été optimisée par OEPP, aurait convergé à un point qui n'est pas son optimum, le P&O permet d'aller chercher l'optimum de l'unité avec une certaine précision. Cependant, si une unité converge très loin de son optimum avec l'OEPP, le P&O prendra plus de temps à trouver l'optimum. Donc, d'autres additions à l'algorithme ont été faites pour accélérer davantage la convergence de l'OEPPC.

Mémorisation de la meilleure position de chaque unité depuis la dernière perturbation

L'ajout de mémoire à l'algorithme a permis de prendre avantage de la nature stochastique de l'OEP. En effet, lors de l'optimisation par OEPP, les unités font une recherche aléatoire dans leur espace de recherche respectif. Il est donc possible qu'une particule de l'OEPP passe dans la région où se trouve son optimum sans qu'elle n'y converge. En rajoutant une variable D_{opt} pour chaque unité qui contient la meilleure position obtenue pour cette unité depuis la dernière perturbation, on peut accélérer l'optimisation par P&O des unités. À cet effet, lorsqu'une particule de l'OEPP a convergé, elle est retirée de son essaim, placée à son D_{opt} , puis l'optimisation par P&O est démarrée sur cette unité.

Amplitude de perturbation variable dans l'algorithme de P&O

Afin d'accélérer la convergence du P&O, un Δ plus grand est utilisé au début de l'optimisation par P&O, soit Δ_1 , puis un Δ petit est utilisé pour affiner la position finale de l'unité, soit Δ_2 . Lorsqu'une unité optimisée par P&O est jugée être en régime permanent (avec une méthode détaillée à la section 2.4.1.1), on la place à son D_{opt} , qui a été mis à jour pendant l'optimisation par P&O, et elle n'en bouge plus. L'unité devient alors une sentinelle pour permettre de détecter une perturbation qui affecterait sa courbe de puissance.

Ajout d'un classificateur pour regrouper les unités présentant des courbes statiques similaires dans l'approche OEPP

Un classificateur a été ajouté à l'algorithme, de façon à permettre le regroupement des unités dont la position de l'optimum dans l'hyperespace est identique ou semblable lorsqu'une perturbation est détectée. Comme il a été expliqué à la section 2.3, les performances de l'OEPP diminuent lorsque les unités à optimiser présentent des différences au niveau de leur position optimale dans l'hyperespace. Le classificateur permet de regrouper les unités semblables dans un même sous-algorithme d'OEPP dans le but que toutes les unités du sous-algorithme convergent à un point qui est proche de leur optimum respectif. Donc, lorsqu'une perturbation survient, une classification des unités perturbées est faite et les groupes créés par le classificateur sont subdivisés en d'autres sous-algorithmes d'OEPP. Par exemple, admettons qu'un système soit composé de 10 unités, dont les unités 1 à 5 ont un $D_{opt} = 100 \Omega$ et les unités 6 à 10 ont un $D_{opt} = 200 \Omega$, les unités 1 à 5 seraient placées dans un sous-algorithme d'OEPP et les unités 6 à 10 dans un autre. De cette façon, les 10 unités convergeraient toutes à leur optimum respectif. Notre algorithme de classification est expliqué plus en détails à la section 2.4.1.3.

Pour résumer le tout, la Figure 2.3 montre toutes les étapes par lesquelles une unité optimisée par OEPPC peut passer. L'OEPPC optimise d'abord toutes les unités par OEPP afin qu'elles convergent rapidement vers une région proche de leur optimum. Lorsqu'une unité a convergé, on place l'unité à sa meilleure position obtenue pendant la recherche depuis la dernière perturbation (D_{opt}). L'unité est ensuite optimisée par P&O avec $\Delta = \Delta_1$. Lorsque la convergence est atteinte, on change le Δ du P&O à $\Delta = \Delta_2$. Lorsque la convergence est de nouveau atteinte, l'unité est placée à D_{opt} et y reste. Il est à noter que la variable D_{opt} est mise à jour à chaque itération de l'algorithme, et aussi que cette variable est réinitialisée lors d'une perturbation sur une unité. En outre, lorsqu'une unité passe du mode OEPP au mode P&O, son D_{opt} est aussi réinitialisé. En procédant de la sorte, on s'assure que le D_{opt} de l'unité ne sera pas influencé par du bruit de mesure qui aurait pu être présent pendant l'optimisation par OEPP. Finalement, si une perturbation survient, une classification est faite

sur les unités perturbées pour les regrouper en fonction de leur résistance optimale D_{opt} , et les unités semblables sont optimisées par OEPP. À noter que chaque groupe créé par le classificateur est optimisé par un sous-algorithme d'OEPP distinct.

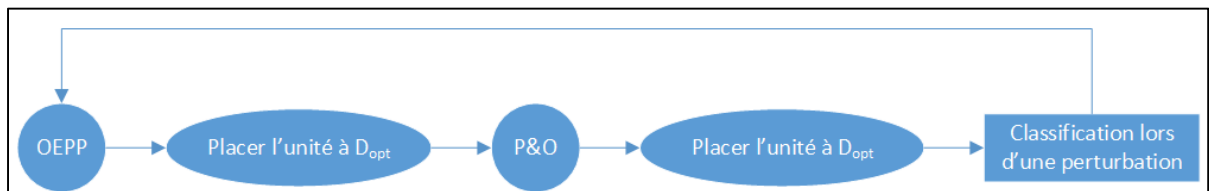


Figure 2.3 Étapes par lesquelles une unité optimisée par OEPPC peut passer

Les prochaines sections présentent plus en détails chaque élément de l'OEPPC, puis l'implémentation de tous ces éléments est élaborée.

2.4.1 Modifications à l'OEPP pour obtenir l'OEPPC

Avant d'aborder plus en détails chaque élément de l'OEPPC, il faut auparavant comprendre que, lors d'une itération d'OEPPC, les algorithmes d'OEPP et de P&O peuvent opérer en même temps, étant donné que les unités sont optimisées d'un point de vue individuel. À des fins de simplification, ces algorithmes sont référés comme étant des sous-algorithmes (SA). L'OEPPC est donc constituée de SA d'OEPP et de SA de P&O.

Les différents SA de l'OEPPC opèrent de façon indépendante. Les unités du système à optimiser sont dynamiquement placées dans différents SA lorsque certaines conditions sont remplies. Donc, il est possible qu'il y ait plusieurs SA d'OEPP qui opèrent et qu'aucun SA de P&O n'opère, ou qu'un seul SA d'OEPP opère ainsi que plusieurs SA de P&O, ou toute autre combinaison de ces sous-algorithmes. Il est à noter qu'une unité ne peut être optimisée par deux SA en même temps. Ainsi, le nombre total de SA est toujours inférieur ou égal au nombre d'unités à optimiser, mais toujours supérieur à zéro. En outre, le fait de procéder de la sorte offre un avantage dans le cas où une perturbation n'affectant que quelques unités du système survenait. En effet, dans un tel cas, l'OEPP redémarrerait une recherche d'optimum

pour toutes les unités, alors qu'il n'aurait été nécessaire de redémarrer une recherche que pour les unités affectées par la perturbation. Ainsi, l'OEPPC présente un avantage supplémentaire par rapport à l'OEPP dans un tel cas car une unité à son optimum y reste à moins qu'une perturbation ne soit détectée sur l'unité.

La Figure 2.4 donne l'ordinogramme de l'OEPPC d'un point de vue haut niveau. L'OEPPC s'effectue séquentiellement comme suit, avec les étapes 2 à 6 représentant une itération de l'algorithme et répétées indéfiniment, et chaque itération est espacée d'un délai d'attente T représentant le temps nécessaire aux unités pour se stabiliser après avoir changé leur point d'opération :

1. Initialiser le système en calculant les positions dans l'hyperespace des unités et positionner ces dernières;
2. Attendre T secondes, puis évaluer les fonctions objectif des unités;
3. Mettre à jour D_{opt} . À cette étape, les valeurs de fitness courantes des unités sont comparées à celles encapsulées par les différents D_{opt} , et les D_{opt} sont mis à jour;
4. Faire une itération des SA d'OEPP s'il y en a. Ajuster les variables de contrôle des unités;
5. Faire une itération des SA de P&O s'il y en a. Ajuster les variables de contrôle des unités;
6. Phase de repositionnement des unités :
 - a. Si des unités ont été perturbées, les classer et les repositionner dans des SA d'OEPP. Si moins de 3 unités ont été perturbées, les placer dans un SA de P&O. Réinitialiser les D_{opt} des unités perturbées, et ajuster les variables de contrôle des unités affectées;
 - b. Si des unités de l'OEPP ont convergé et doivent être placées dans des SA de P&O, y procéder et ajuster les variables de contrôle des unités affectées. Puis, réinitialiser le D_{opt} des unités affectées, et revenir à l'étape 2.

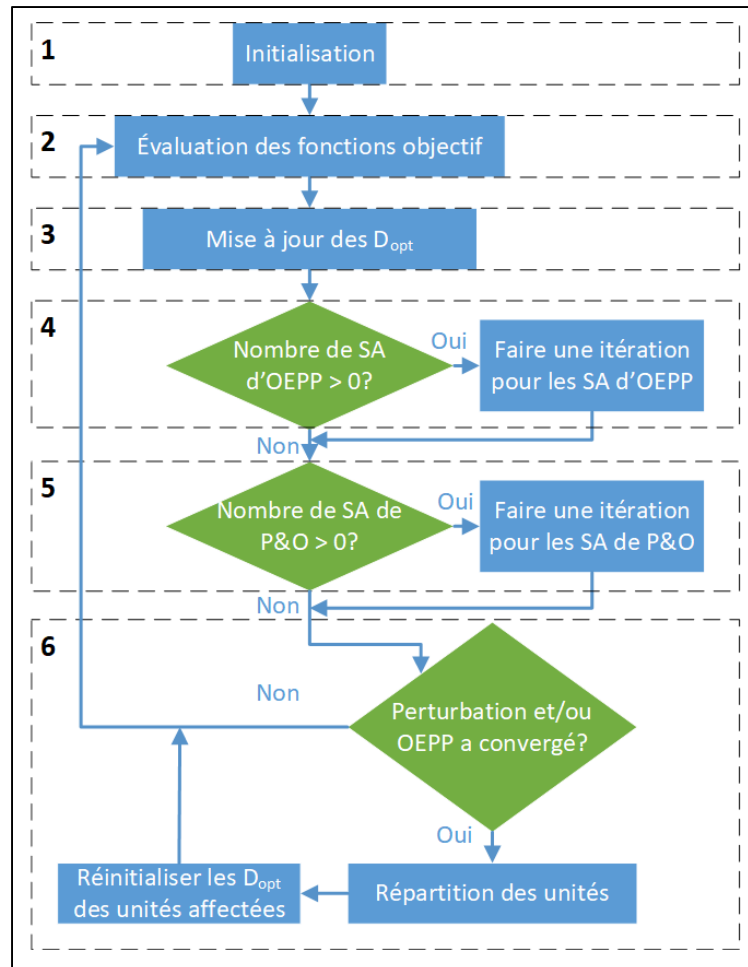


Figure 2.4 Ordinogramme de l'OEPPC

2.4.1.1 Affinage de l'optimum avec P&O

Une méthode hybride combinant l'OEP et le P&O a été déjà développée en 2015 (Sundareswaran, Vignesh kumar et Palani, 2015). Dans cette méthode, l'OEP est appliquée sur un panneau solaire. Lorsque toutes les particules ont convergé, le système passe en mode P&O et raffine la position de la variable de contrôle du panneau PV. Lorsque la différence de la puissance générée d'une itération à une autre est inférieure à une limite ϵ , le système est positionné à la meilleure position mesurée en mode P&O et y reste. Ensuite, lorsqu'une perturbation de puissance est détectée, le système se remet en mode OEP et la recherche d'optimum recommence. L'étude compare expérimentalement les résultats de cette approche

à ceux de l'OEP et du P&O et démontre qu'une plus grande puissance est obtenue par cette méthode et qu'il y a moins d'oscillations en régime permanent.

De la même façon, une unité de l'OEPPC ayant convergé par OEPP est ensuite optimisée par P&O. La différence de l'OEPPC avec la méthode de Sundareswaran, Vignesh Kumar et Palani (2015), en plus de l'utilisation de l'OEPP, est que les unités de l'OEPPC sont évaluées individuellement. Par exemple, si une seule particule d'un SA d'OEPP comprenant 8 particules (i.e. 8 unités) a convergé, l'unité représentée par cette particule est retirée du SA d'OEPP et une optimisation par P&O est appliquée sur celle-ci, réduisant ainsi le nombre de particules du SA d'OEPP initial à sept. En procédant de la sorte, il est possible d'isoler les unités perturbées ensemble, en laissant les unités non perturbées à leur optimum.

En outre, à l'instar de la méthode de Sundareswaran, Vignesh Kumar et Palani (2015), un mécanisme de détection de régime permanent est implémenté afin de diminuer l'amplitude des oscillations en régime permanent dues à la nature du P&O. Cette méthode de détection diffère cependant de celle de Sundareswaran, Vignesh Kumar et Palani. Lorsque le P&O converge, l'amplitude d'oscillation autour de l'optimum sera généralement d'un maximum de $\pm 2\Delta$. Donc, une unité de P&O est assumée être en régime permanent lorsque l'amplitude d'oscillation des n_{voulu} dernières itérations (paramètre à être déterminé par l'utilisateur) est inférieure à $\pm 2\Delta$. Lorsqu'une unité est considérée en régime permanent, son Δ est diminué à Δ_2 , puis l'optimisation continue. Lorsque l'unité est considérée à nouveau être en régime permanent, elle est placée à D_{opt} . Le pseudocode listé ci-dessous décrit les étapes à suivre pour y arriver, où $n_{échantillons}$ représente le nombre d'itérations écoulées depuis le début de l'optimisation par P&O, ou bien le nombre d'itérations depuis la dernière détection de régime permanent. Lorsque l'unité est fixée à D_{opt} , elle n'en bougera plus et la valeur de fitness obtenue à chaque itération sera comparée à celle de l'itération précédente pour déterminer si une perturbation est survenue. Si une perturbation est détectée, l'unité sera retirée du SA de P&O et sera classifiée afin qu'une recherche d'optimum redémarre.

- **SI** $n_{\text{échantillons}} \geq n_{\text{voulu}}$
 - Calculer la moyenne des n_{voulu} dernières itérations (variable *moyenne*)
 - Trouver le maximum des n_{voulu} dernières itérations (variable *max*)
 - Trouver le minimum des n_{voulu} dernières itérations (variable *min*)
 - $lim_{\text{haut}} = max - moyenne$
 - $lim_{\text{bas}} = moyenne - min$
 - **SI** $(abs(lim_{\text{haut}}) \geq 2\Delta) \ || \ (abs(lim_{\text{bas}}) \geq 2\Delta)$
 - L'unité n'est pas en régime permanent, continuer la recherche
 - **SINON**
 - **SI** $\Delta == \Delta_1$

$$\Delta = \Delta_2$$

$$n_{\text{échantillons}} = 0$$
 - **SINON**

L'unité est en régime permanent, la placer à D_{opt} , qui a été mis à jour pendant l'optimisation par P&O

2.4.1.2 Ajout de mémoire

Afin de permettre une convergence plus rapide à l'algorithme, un ajout de mémoire a été implémenté pour profiter de la nature stochastique de l'OEP. Une variable représentant la meilleure position d'une unité, nommée D_{opt} , est ajoutée à l'algorithme. Ainsi, à chaque itération de l'optimisation par OEPPC, la position de chaque unité est comparée à son D_{opt} , et D_{opt} est mis à jour si la valeur de la fonction objectif courante d'une unité est meilleure que celle associée à celle de D_{opt} . D_{opt} est utilisé de trois façons : rapprocher une unité de son optimum lorsqu'elle a convergé avec l'OEPP, positionner une unité à cette position lorsque le P&O oscille autour de l'optimum, et comme critère de regroupement d'unités pour le classificateur. D_{opt} est réinitialisé lorsqu'une perturbation sur l'unité est détectée et lorsque l'unité passe en mode P&O. Le D_{opt} d'une unité j est calculé de la façon suivante :

SI $F_j(i) > F_{D_{opt_j}}$ // Si la valeur de fitness courante est supérieure à celle de D_{opt}
 $F_{D_{opt_j}} = F_j(i)$ // Mettre à jour la valeur de fitness de D_{opt}
 $D_{opt_j} = d_j(i)$ // Mettre à jour la position de D_{opt}

2.4.1.3 Classificateur

Dans leur approche basée sur l'OEP, Chen et al. (2015) ont implémenté un classificateur permettant de regrouper des panneaux solaires ayant des caractéristiques similaires (Chen et al., 2015). Leur méthode de classification est basée sur une estimation du niveau d'ensoleillement reçu par chaque panneau. L'équation (2.1) définit leur méthode de classification, où r_i représente le ratio d'ensoleillement maximal, I_m représente le courant du module PV lorsqu'il est opéré à $0.7 * OCV$, et I_{sc} est le courant de court-circuit du module.

$$r_i = \frac{I_m}{I_{sc}} \quad (2.1)$$

Tirées de Chen et al. (2015)

Leur méthode de classification est intéressante et a présenté de bons résultats, mais elle ne s'applique qu'aux systèmes PV. Qui plus est, leur méthode requiert deux mesures, c'est-à-dire le courant de court-circuit (I_{sc}), et la tension en circuit ouvert (OCV). Le classificateur proposé pour l'OEPPC considère uniquement les positions optimales D_{opt} des unités obtenues pendant l'optimisation. En effet, comme il a été expliqué à la section 2.3, l'OEPP converge bien lorsque l'optimum des fonctions objectif est semblable, même si la puissance optimale diffère. Ainsi, le classificateur regroupera les unités dont la différence entre les positions optimales est inférieure ou égale à une tolérance Δ_c , définie lors de l'implémentation de l'algorithme. En outre, à la suite de nos simulations, il a été déterminé que le nombre minimal de particules nécessaire pour que l'OEP converge efficacement est de trois. Afin de s'assurer que cette condition soit remplie, un groupe créé par le classificateur comptant moins de trois unités sera jumelé au groupe voisin le plus proche, de manière qu'il n'y ait aucun groupe comptant moins de trois unités. Les unités des groupes résultant de la

classification sont placées dans des SA d'OEPP, avec toutes les unités d'un même groupe étant dans un SA d'OEPP distinct. S'il y a moins de 3 unités à classer, il n'y a pas de classification faite et les unités sont placées directement dans un SA de P&O.

Exemple

Considérons un système composé de 10 unités ayant toutes convergé à leur optimum. Les positions optimales respectives dans l'espace de recherche D des unités 1 à 10 sont : [73, 77, 100, 95, 173, 67, 120, 89, 190, 195]. Pour $\Delta_c = 15$, la classification des unités résulterait initialement en 5 groupes, puis les groupes de moins de 3 unités seraient fusionnés pour finalement obtenir 3 groupes, tel qu'illustré à la Figure 2.5.

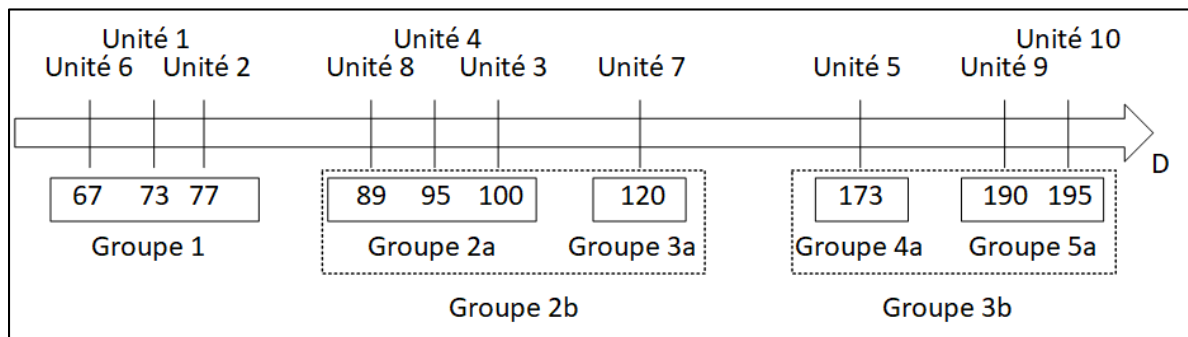


Figure 2.5 Exemple de classification de 10 unités dans un espace de recherche D

Cet exemple aide à visualiser les mécanismes du classificateur. Premièrement, les positions optimales des unités des groupes créés initialement sont toutes à l'intérieur de la tolérance $\Delta_c = 15$ (groupes 1, 2a, 3a, 4a, et 5a). Deuxièmement, on voit que l'unité 2 aurait pu être placée dans les groupes 1 et 2a ($89 - 77 = 12$; $12 \leq \Delta_c = 15$), mais qu'elle a finalement été placée dans le groupe 1, car c'était le groupe dont elle était le plus semblable ($77 - 73 = 4$; $4 < 12 < \Delta_c = 15$). Troisièmement, les groupes ayant moins de 3 unités (groupes 3a, 4a, et 5a) ont été jumelés au groupe dont ils étaient le plus semblable, résultant en deux groupes, soient les groupes 2b et 3b. On remarque que l'unité 5 aurait pu être remplacée dans le groupe 2b, mais qu'elle était plus proche de l'unité 9, ce qui a créé le groupe 3b ($190 - 173 = 17$; $173 - 120 = 53$; $17 < 53$).

La Figure 2.6 montre l'ordinogramme du classificateur de l'OEPPC. À noter qu'un ordinogramme plus détaillé permettant la programmation du classificateur est donné à l'ANNEXE II.

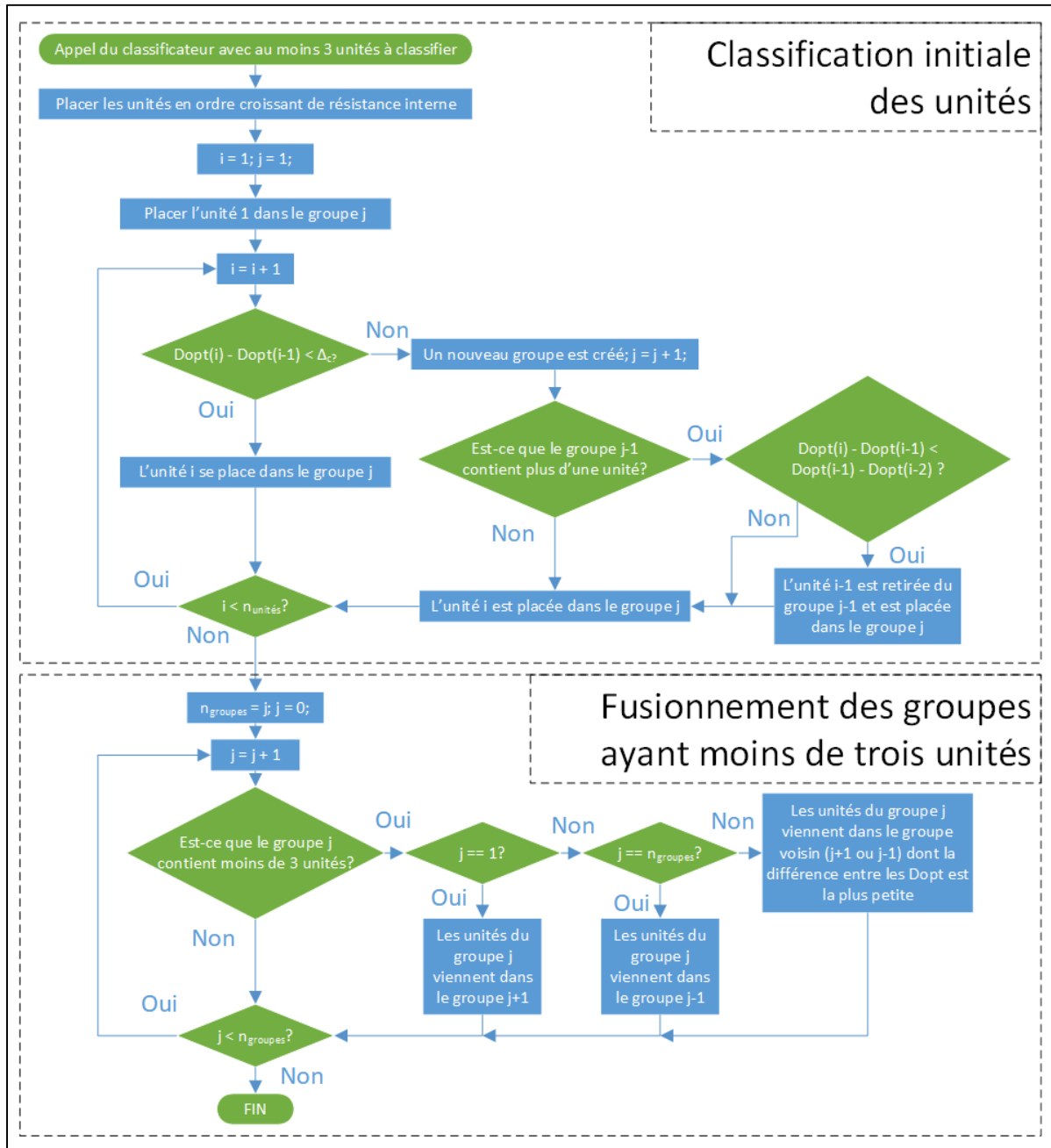


Figure 2.6 Ordinogramme du classificateur

2.4.2 Ordinogrammes de l'OEPPC

Cette sous-section entre plus en détails dans l'OEPPC. Plus spécifiquement, les sous-algorithmes d'OEPP et de P&O sont étudiés. Ensuite, l'algorithme de repositionnement des unités en utilisant le classificateur est expliqué.

2.4.2.1 Sous-algorithmes d'OEPP de l'OEPPC

L'élément d'OEPP de l'OEPPC ressemble grandement à l'algorithme décrit à la section 2.3, à quelques exceptions près. Notamment, une intelligence pour les particules perturbées et les particules ayant convergé a été ajoutée. La détection de perturbations est faite tel que décrit à la section 2.2.3, mais la méthode d'adaptation à celles-ci est différente. Lorsqu'une particule de l'OEPP détecte une perturbation, au lieu de réinitialiser l'essaim au complet, cette particule est retirée du SA d'OEPP et l'unité qui y est associée est ensuite classifiée pour être placée dans un SA différent. En effet, si seulement quelques particules d'un essaim ont été perturbées, il serait inutile, voire même désavantageux, de redémarrer une recherche pour tout l'essaim. En outre, lorsqu'une particule a convergé, l'unité associée à la particule est retirée du SA d'OEPP et est repositionnée à la fin de l'itération d'OEPPC dans un SA de P&O. Ces nuances avec l'OEP sont représentées à la Figure 2.7 par les éléments « Calcul des prochaine positions », « Particules convergées? », et « Particules perturbées? ».

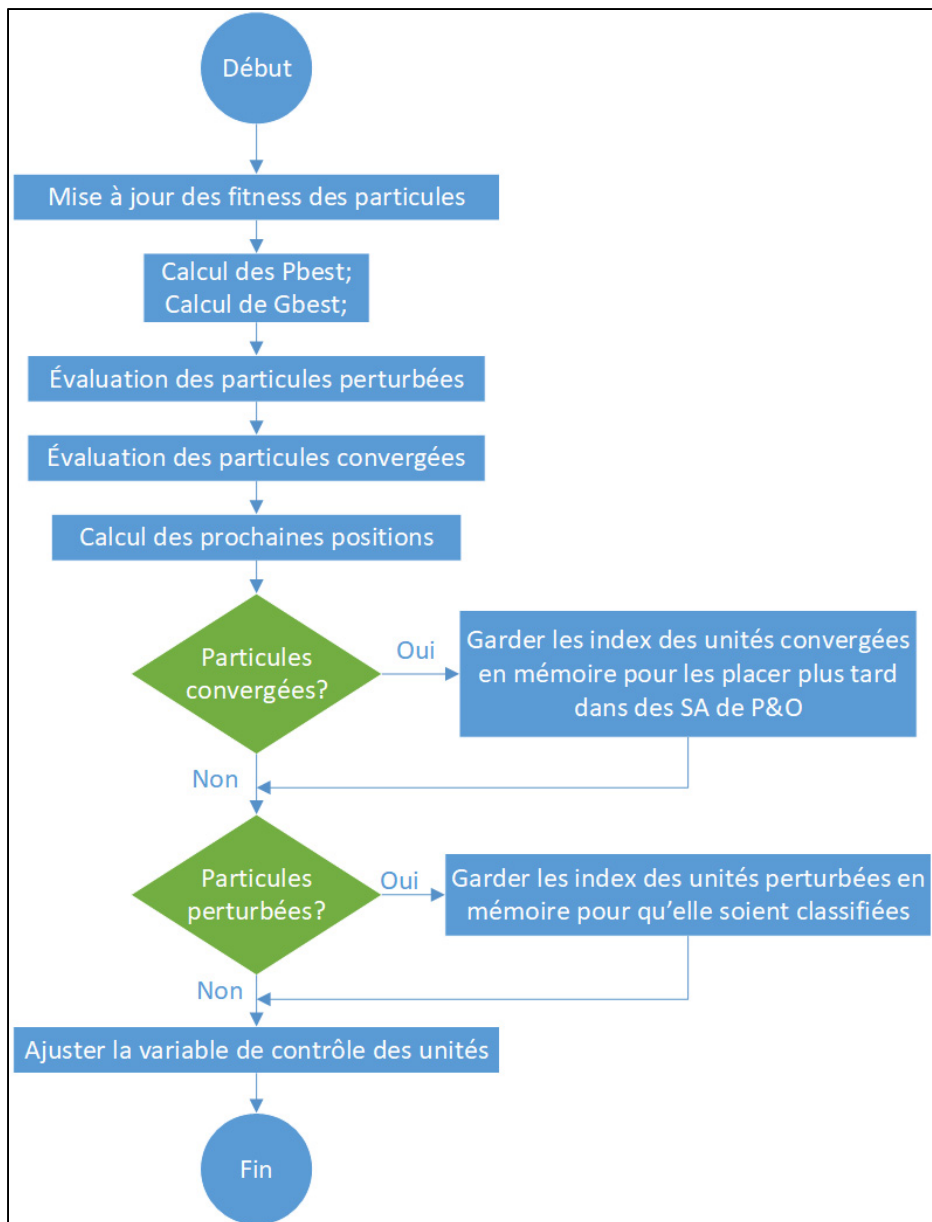


Figure 2.7 Ordinogramme de l'OEPP de l'OEPPC

2.4.2.2 Sous-algorithme de P&O de l'OEPPC

L'ordinogramme de la partie P&O de l'OEPPC est présenté à la Figure 2.8, où $n_{unités}$ représente le nombre d'unités d'un SA de P&O. Les valeurs de fitness des unités sont mises à jour, puis la convergence des unités est évaluée tel que décrit à la section 2.4.1.1. Si une unité est en régime permanent et que $\Delta == \Delta_1$, le paramètre Δ est fixé à Δ_2 , et la nouvelle position

calculée avec l'équation (1.7). Si l'unité est en régime permanent, mais que $\Delta == \Delta_2$, l'unité est placée à son D_{opt} , qui a été mis à jour à chaque itération, et n'en bouge plus. Si l'unité n'est pas en régime permanent, le calcul de la position d'une unité est fait tel que décrit à la section 1.3.2 avec l'équation (1.7). Comme les unités convergées restent au même point, il est facile de déterminer si une perturbation est survenue, car il suffit de comparer la valeur de fitness courante à la valeur de fitness précédente, de la même façon que les particules sentinelles de l'OEPP. Si une unité ayant convergé détecte une perturbation, cette unité est retirée du SA de P&O pour être classifiée et repositionnée plus tard dans la même itération d'OEPPC. La variable de contrôle des unités est ensuite ajustée.

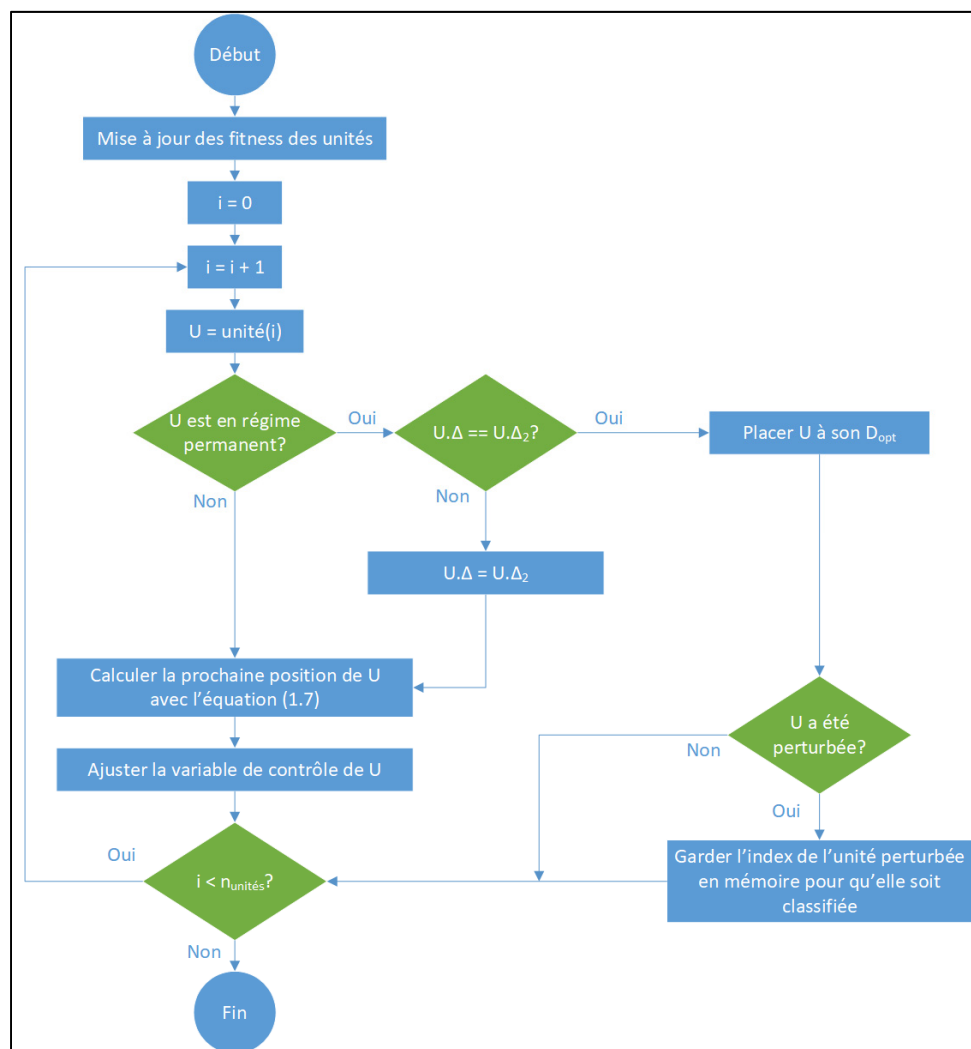


Figure 2.8 Ordinogramme du P&O de l'OEPPC

2.4.2.3 Repositionnement des unités de l'OEPPC

La Figure 2.9 représente la méthode de repositionnement des unités. Les index des unités perturbées de l'OEPP et du P&O sont placés dans un vecteur, et ce vecteur est passé au classificateur afin de regrouper les unités semblables ensemble, tel que décrit à la section 2.4.1.3. C'est à cette étape que les unités sont vraiment retirées de leur sous-algorithme respectif et placées dans des SA d'OEPP. En outre, les unités d'OEPP devant être placées dans un SA de P&O le sont à cette étape, en plus d'être placées préalablement à leur D_{opt} .

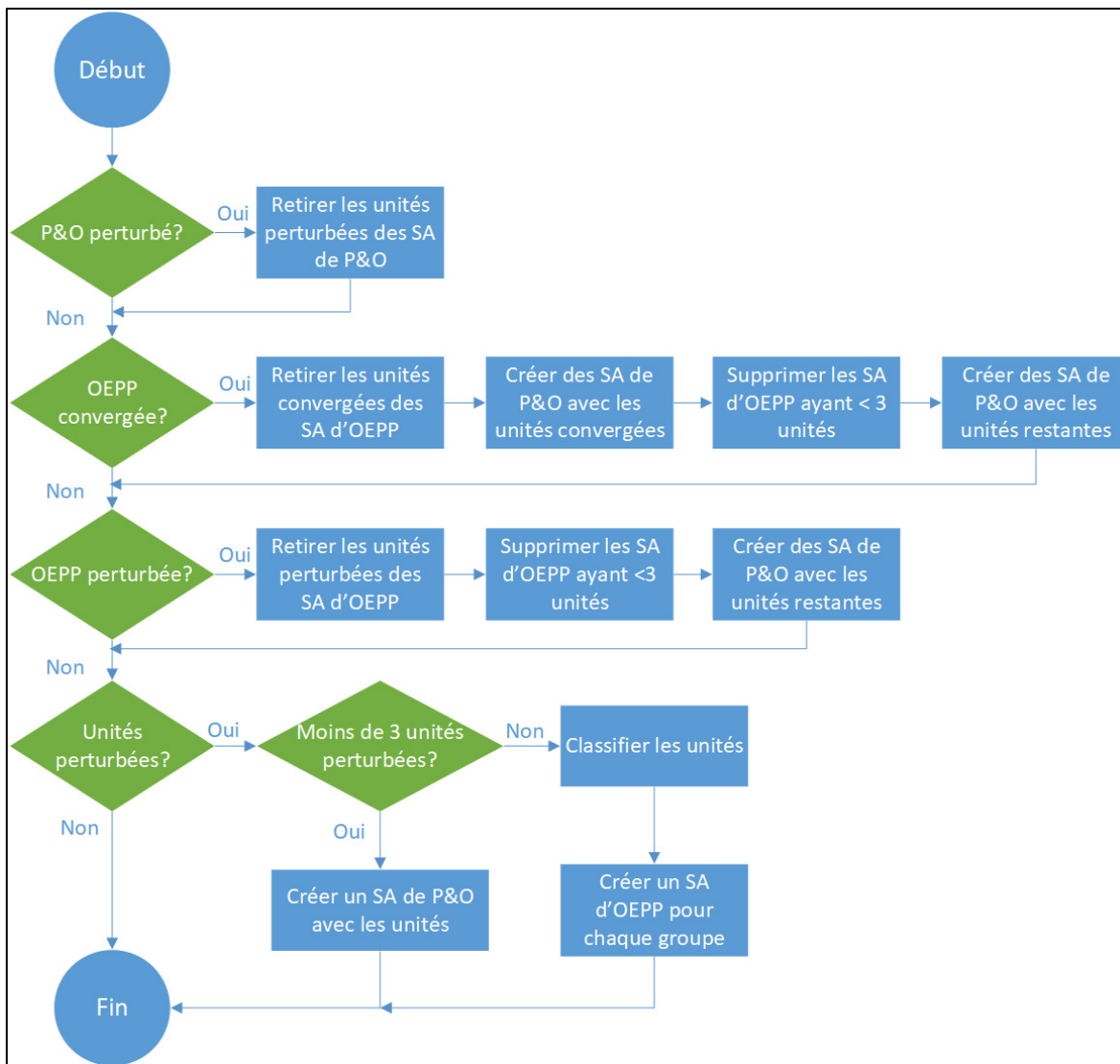


Figure 2.9 Ordinoigramme de l'algorithme de repositionnement des unités

Les SA d'OEPP demandent une attention particulière car, dans le cas où un de ces SA compte moins de trois particules après avoir retiré les unités perturbées ou convergées, la condition minimale de trois particules dans un essaim d'OEPP n'est plus respectée. Ainsi, les SA d'OEPP comptant moins de trois particules sont supprimés, et les unités qui restaient dans ces essais sont placées dans un SA de P&O, après avoir été placée à leur D_{opt} . En outre, si le nombre total d'unités perturbées est inférieur à 3, aucune classification n'est faite et les unités perturbées sont placées directement dans un SA de P&O.

2.5 Conclusion

Dans ce chapitre, trois méthodes basées sur l'OEP ont été élaborées. D'abord, l'OEP a été présentée avec quelques modifications nécessaires pour l'optimisation en temps réel de la puissance de sortie d'unités dynamiques, devenant ainsi l'optimisation par essais particuliers séquentielle (OEPS). Ensuite, l'optimisation par essais particuliers parallélisée (OEPP) a été expliquée. Cet algorithme est avantageux lorsqu'il est nécessaire d'optimiser plusieurs unités présentant des courbes statiques de puissance identiques ou semblables. Cependant, cet algorithme montre quelques lacunes lorsque ces courbes diffèrent trop, spécialement lorsque les résistances internes des unités sont trop différentes. Ainsi, l'optimisation par essais particuliers avec classificateur (OEPPC) a ensuite été proposée. Cet algorithme tente de combler les lacunes de l'OEPP tout en conservant son avantage de vitesse. L'OEPPC intègre des éléments de l'optimisation par P&O afin d'affiner la position d'unités ayant convergé par OEPP. En outre, une méthode de classification permettant de regrouper les unités semblables dans des essais particuliers séparés lors des perturbations a été détaillée.

Le CHAPITRE 3 compare les performances de ces algorithmes sur un système basé sur des PCM à celle du P&O, et le CHAPITRE 4 fait de même sur un système basé sur des cellules PV.

CHAPITRE 3

OPTIMISATION EN TEMPS RÉEL DE LA PUISSANCE DE SORTIE DE PILES À COMBUSTIBLE MICROBIENNES

3.1 Introduction

Peu d'études ont été faites par rapport à l'optimisation en temps réel de la puissance de sortie de piles à combustibles microbiennes. Dans les études qui touchent à ce sujet, l'algorithme d'optimisation choisi est souvent le P&O (Woodward et al., 2010) en raison de sa simplicité. À notre connaissance, l'OEP n'a encore jamais été appliquée à ce contexte. Ce chapitre a donc comme premier objectif d'évaluer l'applicabilité des méthodes basées sur l'OEP pour l'OTR de la puissance de sortie des piles à combustibles microbiennes. Un second objectif est de comparer les performances des algorithmes P&O, OEPS, OEPP, et OEPPC afin de démontrer les avantages qu'offre l'OEPPC par rapport aux autres.

Les systèmes étudiés sont composés de 10 piles à combustible microbiennes en simulations et de 3 PCM en expérimentations.

3.2 Simulation des approches P&O, OEPS, OEPP, et OEPPC

3.2.1 Procédure expérimentale

Les simulations sur des PCM ont été faites à l'aide du modèle de Pinto et al. (2010) présenté à la section 1.2.2. Ce modèle fondamental permet de caractériser la dynamique interne d'une PCM et d'établir un contexte de simulation où l'on peut dicter les différences entre les PCM et gérer les perturbations de l'environnement par l'ajustement de certains de ses paramètres. Les simulations ont été faites dans l'environnement MATLAB/Simulink. L'implémentation du modèle est présentée à l'ANNEXE I, au sous-titre « Modèle d'origine ». Le modèle utilise les variables listées ci-dessous :

1. S : la concentration massique du substrat à l'intérieur de la pile;
2. x_a : la concentration massique de microorganismes anodophiles;
3. x_m : la concentration massique de microorganismes méthanogènes;
4. M_{ox} : le rapport de médiateur oxydé par microorganisme anodophile;
5. S_0 : la concentration massique du substrat d'alimentation;
6. R_{ext} : la résistance externe reliant les deux électrodes de la pile.

Il est à noter que les variables 1 à 4 représentent les variables d'état du système dont les valeurs sont mises à jour par les équations d'état non linéaires du modèle, et que les variables 5 et 6 sont les valeurs d'entrée du modèle, contrôlées par l'utilisateur. La puissance de sortie de la pile est calculée et donnée en sortie. La Figure 3.1 montre un exemple graphique de l'utilisation du modèle sous Simulink.

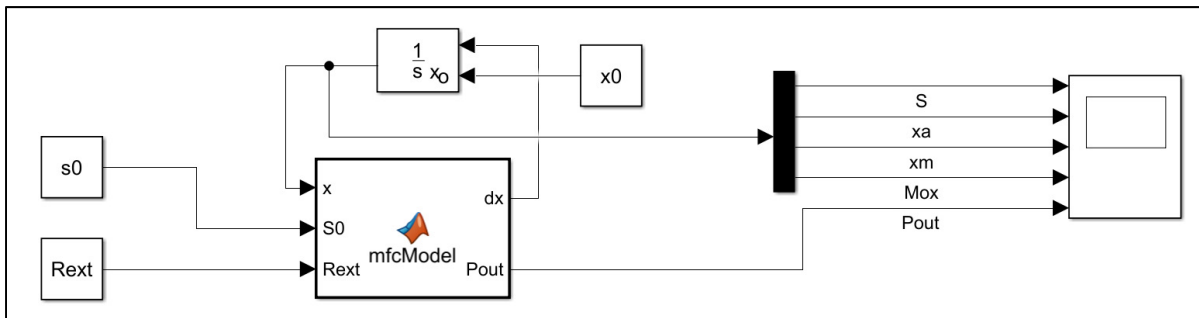


Figure 3.1 Exemple graphique de l'implémentation du modèle d'une PCM de Pinto et al. (2010)

En considérant les concentrations des microorganismes anodophiles et méthanogènes constantes, il est possible d'accélérer le temps de simulation de ce modèle. Ainsi, on ne considère que la dynamique plus rapide de la PCM et on néglige la dynamique très lente associée à l'évolution des microorganismes de la pile. En considérant x_a et x_m comme étant constantes, on réduit le nombre de variables dynamiques du modèle à deux, c'est-à-dire S et M_{ox} . L'implémentation en MATLAB de ce modèle accéléré est définie à l'ANNEXE I, au sous-titre « Modèle accéléré ».

Une des variables qui permet de simuler des perturbations sur une PCM est la concentration massique de substrat d'alimentation S_0 . Cette variable d'entrée au modèle influence directement la dynamique interne de la PCM et est utilisée dans ce chapitre pour simuler une perturbation environnementale. Elle est aussi utilisée pour simuler les différences entre plusieurs PCM. Par exemple, une PCM pourrait avoir un $S_0 = 700 \text{ mg/L}$ et une autre pourrait avoir un $S_0 = 300 \text{ mg/L}$, ce qui résulterait en des courbes de puissance statiques différentes pour les deux PCM, car leur optimum respectif serait conséquemment différent.

La variable de contrôle des PCM est R_{ext} , qui dicte la charge connectée à une PCM en Ohms. En faisant varier cette charge et en attendant que le système se stabilise à chaque point, on peut obtenir la courbe de puissance statique d'une PCM. La Figure 3.2 montre la courbe de puissance statique de deux PCM ayant des S_0 respectifs de 700 mg/L et de 300 mg/L. Ces courbes ont été obtenues avec le modèle de Pinto et al. (2010) accéléré en faisant varier la charge connectée aux PCM et en attendant 2.4 h entre chaque point d'opération. Comme on peut l'observer, une variation d' S_0 occasionne un changement de la courbe de puissance d'une PCM, et une variation d' R_{ext} occasionne un changement de sa puissance sur une même courbe statique. Les simulations algorithmiques se font selon les étapes suivantes :

1. Initialiser les variables d'état du système et déterminer les résistances externes initiales des PCM;
2. Définir le S_0 de chaque PCM. C'est à cette étape que l'on peut induire une perturbation sur une PCM;
3. Positionner les R_{ext} des PCM en fonction de ce qui a été calculé par l'algorithme d'optimisation;
4. Intégrer le modèle rapide de chaque PCM pendant 2.4h avec l'intégrateur « ode15s » de MATLAB;
5. Évaluer les puissances des PCM;
6. Calculer les nouvelles positions des unités avec l'algorithme d'optimisation;
7. Revenir à l'étape 2 jusqu'à ce que le nombre d'itérations voulu soit atteint.

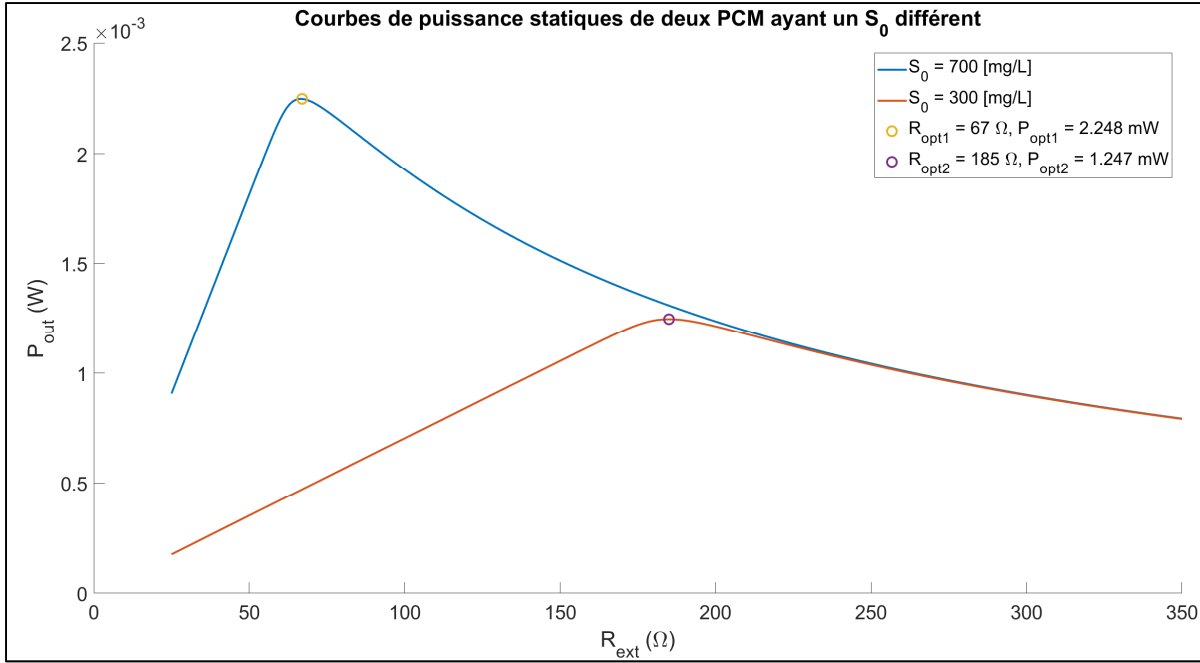


Figure 3.2 Courbes de puissance statiques de deux PCM ayant un S_0 différent

3.2.2 Résultats de simulation

3.2.2.1 Définition des cas d'étude et des critères de performance

Les simulations ont été faites sur quatre algorithmes, soient : le P&O, l'OEPS, l'OEPP, et l'OEPPC. Deux cas d'étude sont considérés, soient : l'optimisation d'unités identiques en présence de perturbations, et l'optimisation d'unités non identiques en présence de perturbations. Il est à noter qu'un troisième cas d'étude a été fait, soit : l'optimisation d'unités non identiques avec perturbations sur quelques unités. Ce cas d'étude est présenté à l'ANNEXE III. Il n'est pas présenté dans ce chapitre, car les résultats obtenus ajoutaient peu d'information par rapport aux résultats obtenus dans les deux premiers cas d'étude.

Les performances des algorithmes sont comparées selon les critères suivants :

1. Le temps de convergence T_{conv} , défini comme étant le temps à partir duquel les valeurs minimale et maximale de la variable de commande d'une unité ne varient que de $\pm 1\%$. Le

- temps de convergence est mesuré en nombre d'itérations pour chaque unité, et une moyenne est faite sur toutes les unités pour obtenir le temps de convergence global d'une simulation;
2. L'énergie totale récoltée en kilojoules, notée E_{total} . L'énergie totale est définie comme étant la somme de l'énergie récoltée de chaque unité pendant l'optimisation;
 3. La puissance de sortie moyenne en mW obtenue à partir du temps de convergence T_{conv} , notée P_{conv} . La puissance de sortie moyenne finale est calculée en faisant la moyenne de la puissance de sortie de chaque unité à partir de l'itération définie par T_{conv} jusqu'à l'itération finale, puis en faisant la somme des puissances moyennes de chaque unité.

Comme le P&O est déterministe, les résultats d'une seule simulation sont présentés. Cependant, en raison de la nature stochastique de l'OEP, 10 simulations sont effectuées pour l'OEPS, l'OEPP, et l'OEPPC, puis les moyennes de ces résultats sont présentées pour chaque cas d'étude.

3.2.2.2 Optimisation d'unités identiques en présence de perturbations

Dans ce cas d'étude, 10 PCM ayant des courbes statiques de puissance identiques sont utilisées. Tout d'abord, les PCM ont une concentration $S_0 = 700 \text{ mg/L}$ pendant 100 itérations (une itération représente 2.4 heures). À la centième itération, une perturbation sur la concentration S_0 de chaque PCM survient sur chaque PCM simultanément, où $S_0 = 300 \text{ mg/L}$. Puis, 100 itérations plus tard, une nouvelle perturbation survient et la concentration S_0 de chaque unité est ramenée à 700 mg/L pendant les 100 prochaines itérations. Au total, les algorithmes optimisent les PCM pendant 300 itérations, ou 720 heures. Les courbes de puissance d'une PCM avec $S_0 = 700 \text{ mg/L}$ et $S_0 = 300 \text{ mg/L}$ sont montrées à la Figure 3.2.

Le Tableau 3.1 donne les valeurs auxquelles les paramètres de l'algorithme P&O ont été fixés lors des simulations. La Figure 3.3 montre les résultats obtenus de l'optimisation de 10 PCM identiques en présence de perturbations par la méthode P&O. Comme on peut le voir,

les 10 courbes sont identiques. Le P&O prend plus de temps à trouver l’optimum pendant les itérations 1 à 100 comparativement aux itérations subséquentes en raison de la valeur initiale qui est plus loin de l’optimum. L’algorithme est ensuite plus rapide par la suite étant donné que le Δ utilisé est assez agressif. Cependant, comme il a été expliqué à la section 1.3.2, un Δ plus grand signifie aussi une oscillation en régime permanent de plus grande amplitude, comme on peut le voir à la Figure 3.3. Les résultats de cette simulation sont présentés au Tableau 3.5.

Tableau 3.1 Paramètres de P&O utilisés pour les simulations

Paramètre	Valeur	Paramètre	Valeur
Δ	15 Ω	Position minimale	10 Ω
Position initiale	200 Ω	Position maximale	350 Ω

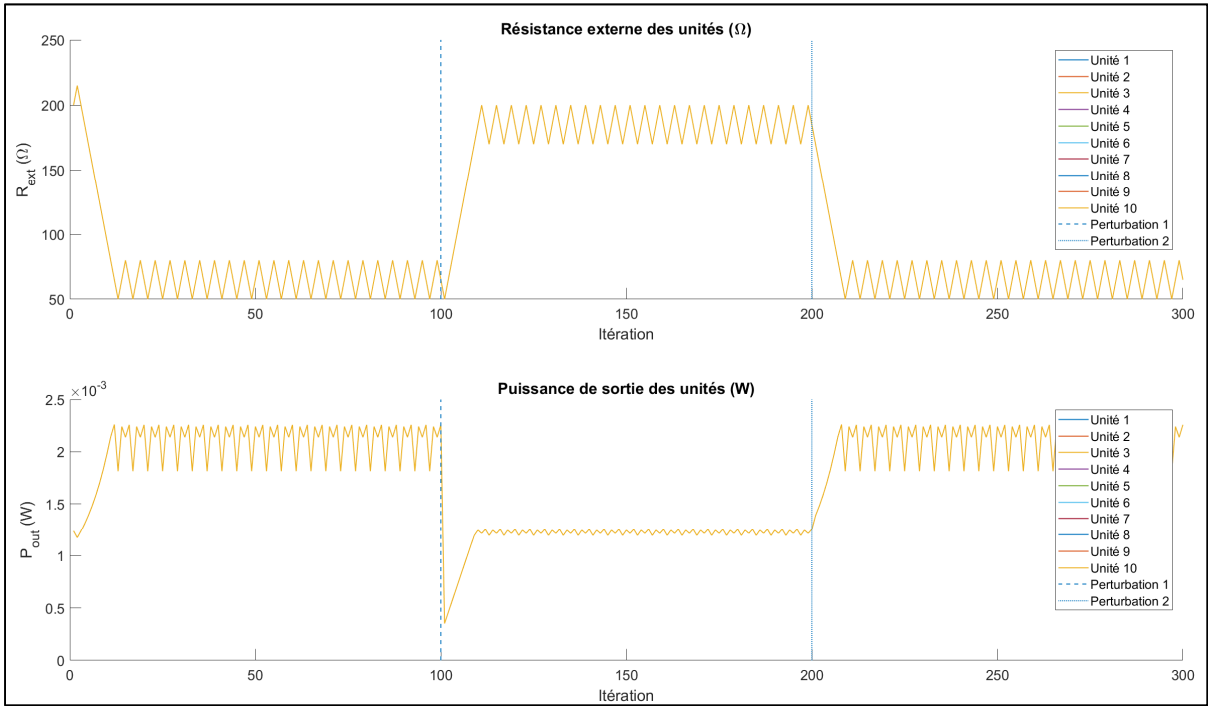


Figure 3.3 Optimisation par P&O de PCM identiques en présence de perturbations

Le Tableau 3.2 donne les paramètres utilisés pour les simulations d'OEPS sur le système. Ces paramètres sont utilisés pour chaque cas d'étude de l'OEPS de cette section. Il est à noter que ces paramètres ont été choisis en faisant plusieurs simulations préliminaires, jusqu'à ce que des paramètres offrant des performances acceptables soient obtenus. À la section 2.4, il a été mentionné que le nombre minimal de particules pour que l'OEP converge bien est de 3. Cependant, comme il est noté au Tableau 3.2, le nombre de particules de l'OEPS est fixé à 5. Cette décision a été prise car, bien que 3 particules soient suffisantes pour l'OEPP et l'OEPPC, l'OEPS convergeait quelques fois à un point qui n'était pas l'optimum. Cela était dû au positionnement initial aléatoire des particules, qui faisait parfois en sorte que les particules n'exploraient pas la région où se trouve l'optimum de la fonction objectif. En augmentant le nombre de particules à 5, le temps de convergence de l'OEPS a été augmenté, mais la précision de l'algorithme a été améliorée. La Figure 3.4 donne un exemple d'une optimisation par OEPS de 10 PCM identiques en présence de perturbations. Comme on peut l'observer, l'OEPS converge beaucoup plus lentement que le P&O, mais l'OEPS ne présente aucune oscillation en régime permanent. Ainsi, pour une même durée en régime permanent, l'OEPS récoltera plus d'énergie que le P&O. Les résultats moyennés de 10 simulations sont présentés au Tableau 3.5.

Tableau 3.2 Paramètres de l'OEPS utilisés pour les simulations

Paramètre	Valeur	Paramètre	Valeur
c_1	0.5	Marge des sentinelles	$\pm 3\%$
c_2	1.2	Position minimale	10 Ω
ω	0.3	Position maximale	350 Ω
Nombre de particules	5		

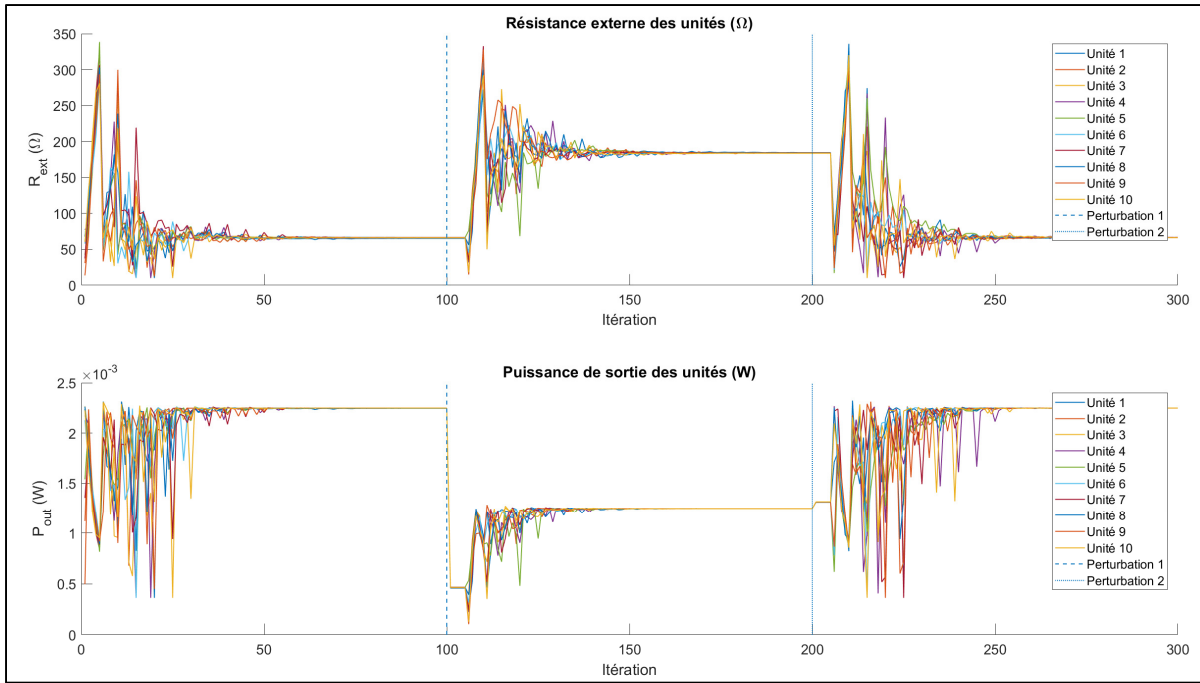


Figure 3.4 Optimisation par OEPS de PCM identiques en présence de perturbations

Le Tableau 3.3 donne les paramètres utilisés pour les simulations de l'OEPP. Ces paramètres sont utilisés pour chaque cas d'étude de l'OEPP de cette section. Les coefficients de l'algorithme sont les mêmes que pour l'OEPS, mais le nombre de particules est fixé à 10, soit le nombre d'unités à optimiser. En effet, comme le système comporte 10 unités, chaque unité est traitée comme une particule, ce qui résulte en un essaim de 10 particules. La Figure 3.5 montre un exemple d'optimisation par OEPP de 10 PCM identiques en présence de perturbations. Il est clair que l'OEPP converge beaucoup plus rapidement que le P&O et que l'OEPS. De plus, étant donné que les unités sont identiques, la précision de l'OEPP est la même que celle de l'OEPS. Ainsi, pour un même temps d'optimisation, l'OEPP récoltera plus d'énergie que l'OEPS car l'algorithme converge beaucoup plus rapidement. Les résultats moyennés de 10 simulations sont présentés au Tableau 3.5.

Tableau 3.3 Paramètres de l'OEPP utilisés pour les simulations

Paramètre	Valeur	Paramètre	Valeur
c_1	0.5	Marge des sentinelles	$\pm 3\%$
c_2	1.2	Position minimale	10Ω
ω	0.3	Position maximale	350Ω
Nombre de particules	10		

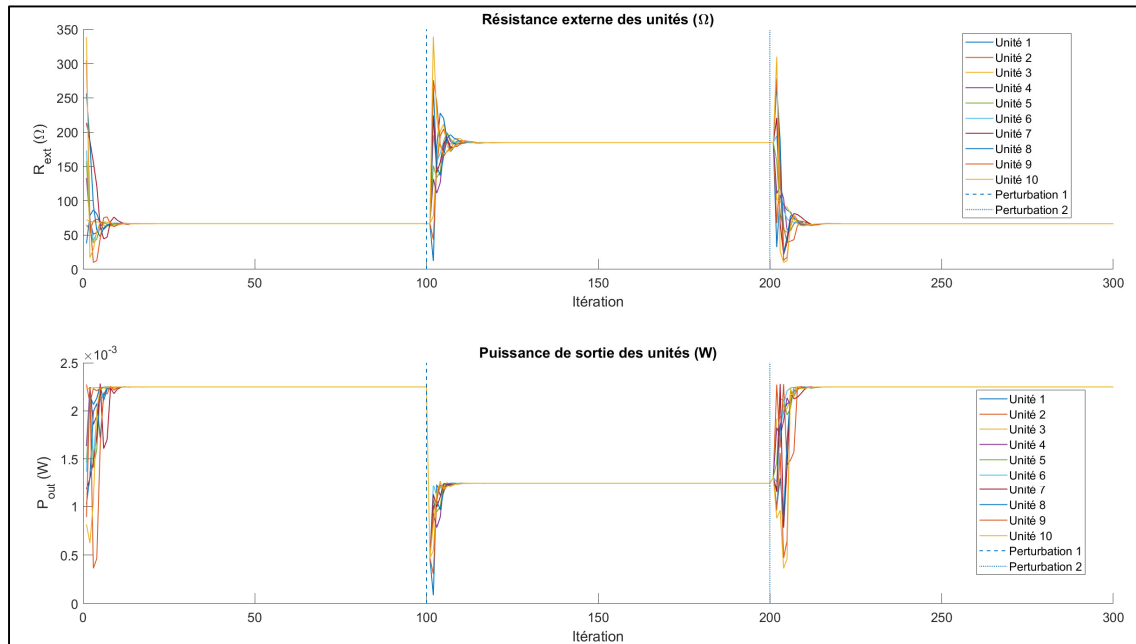


Figure 3.5 Optimisation par OEPP de PCM identiques en présence de perturbations

Le Tableau 3.4 donne les paramètres utilisés pour les simulations d'OEPPC sur le système. Ces paramètres sont utilisés pour chaque cas d'étude de l'OEPPC de cette section. Les coefficients c_1 , c_2 , et ω sont les mêmes que pour l'OEPS et l'OEPP. Le nombre minimal de particules indique la taille minimale d'un groupe créé par le classificateur. Le paramètre Δ_c est en fait la marge du classificateur, telle que décrite à la section 2.4.1.3. Le régime permanent est atteint pour une particule d'un SA d'OEPP lorsque sa position ne varie que de $\pm 1\%$ pendant 5 itérations. Le régime permanent est atteint pour une unité d'un SA de P&O lorsque sa position ne varie que de $\pm 2\Delta$ pendant 7 itérations. Les unités de l'OEPPC

optimisées par P&O ont initialement un $\Delta = \Delta_1 = 5 \Omega$ puis, lorsque la première convergence est atteinte, Δ devient égal à $\Delta_2 = 1 \Omega$.

Tableau 3.4 Paramètres de l'OEPPC utilisés pour les simulations

Paramètre	Valeur	Paramètre	Valeur
c_1	0.5	Δ_c	$\pm 10 \Omega$
c_2	1.2	Position minimale	10Ω
ω	0.3	Position maximale	350Ω
Nombre minimal de particules	3	Δ_1 (P&O)	5Ω
Marge des sentinelles	$\pm 3\%$	Δ_2 (P&O)	1Ω
Nombre d'itérations définissant la convergence pour l'OEPP	5	Amplitude de convergence pour l'OEPP	$\pm 1\%$
Nombre d'itérations définissant la convergence pour le P&O	7	Amplitude de convergence pour le P&O	$\pm 2\Delta$

La Figure 3.6 donne un exemple graphique d'une optimisation par OEPPC de 10 PCM identiques en présence de perturbations. Comme on peut le voir, vu que les unités sont identiques, l'OEPPC se comporte de la même façon que l'OEPP, à l'exception de l'affinement des positions finales par P&O. Cet affinage d'optimum peut être observé sur la Figure 3.6 par les oscillations après que les particules aient convergé. Les résultats moyennés de 10 simulations sont présentés au Tableau 3.5. Il est à noter que l'affinage par P&O n'a pas d'avantage clair dans ce cas d'étude, car les unités avaient déjà convergé à leur optimum par l'optimisation OEPP. Cependant, dans le cas où les unités auraient eu des courbes de puissance différentes, l'OEPP n'aurait pas réussi à amener toutes les unités à leur optimum respectif. Il est donc nécessaire d'utiliser l'affinage par P&O pour assurer la convergence optimale des unités.

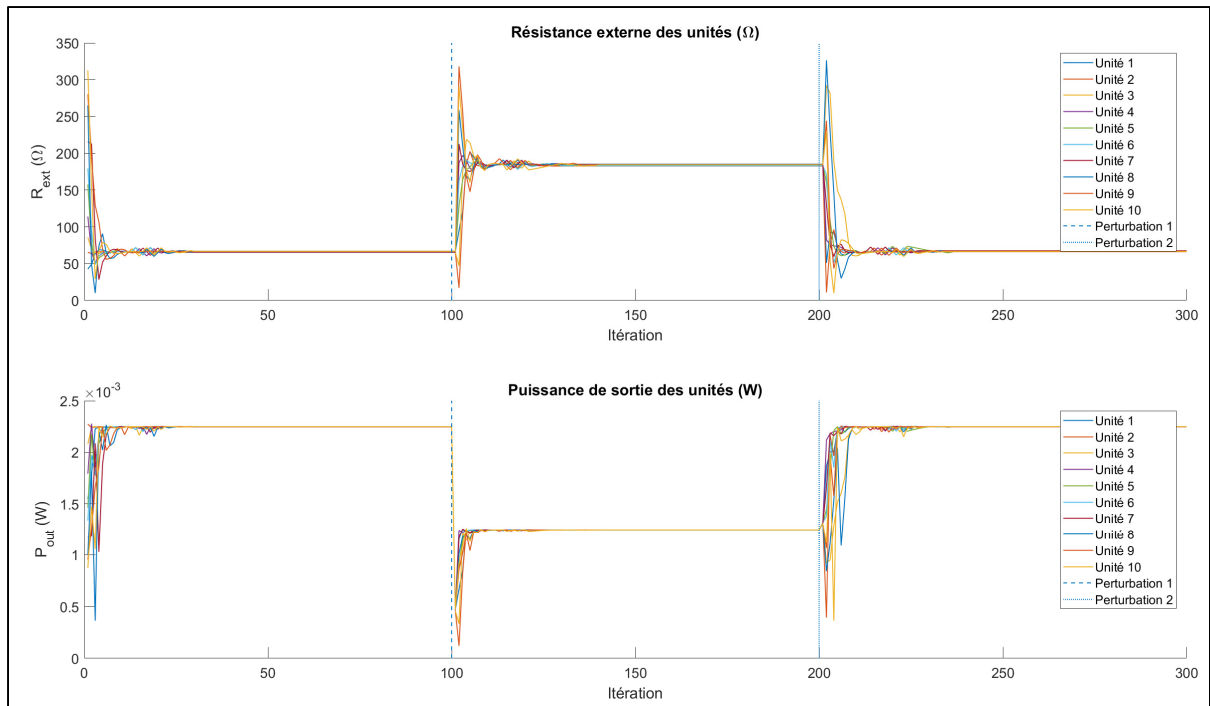


Figure 3.6 Optimisation par OEPPC de PCM identiques en présence de perturbations

Le Tableau 3.5 montre les résultats de simulation pour les quatre algorithmes étudiés. Le tableau est divisé en 3 sections, c'est-à-dire les itérations 1 à 100, les itérations 101 à 200, et les itérations 201 à 300. Les itérations 1 à 100 représentent l'optimisation initiale du système sans perturbations. À l'itération 100, une perturbation est appliquée au système, puis une deuxième perturbation est appliquée à l'itération 200. Les meilleures performances ont été mises en gras et en italique.

À prime abord, on note que le P&O a collecté moins d'énergie pendant les itérations 1 à 100. Cela s'explique par sa position initiale de 200 Ω , qui se trouvait plus loin de l'optimum des unités, donc l'algorithme a pris plus de temps à trouver l'optimum. On note aussi que l'OEPP et l'OEPPC sont tous deux meilleurs que l'OEPS, avec une performance légèrement meilleure pour l'OEPP. Cela est tout à fait normal, étant donné que pour des unités identiques, l'OEPPC se comporte de la même façon que l'OEPP, mais que l'affinement des positions finales par P&O a été ajouté. Ainsi, cet ajout diminue quelque peu l'efficacité de l'algorithme dans ce cas d'étude spécifique, mais de très peu.

Tableau 3.5 Résultats de simulation de PCM identiques en présence de perturbations

Itérations	Critère	P&O	OEPS	OEPP	OEPPC
[1, 100] ($S_0 = 700$)	E_{total} (kJ)	17.586	18.096	19.051	19.035
	T_{conv} (itérations)	12.00	63.71	10.45	23.12
	P_{conv} (mW)	21.149	22.486	22.488	22.485
[101, 200] ($S_0 = 300$)	E_{total} (kJ)	10.172	9.948	10.552	10.565
	T_{conv} (itérations)	10.00	57.49	10.16	21.33
	P_{conv} (mW)	12.260	12.469	12.474	12.470
[201, 300] ($S_0 = 700$)	E_{total} (kJ)	17.881	17.650	18.944	18.956
	T_{conv} (itérations)	8.00	68.20	12.29	23.12
	P_{conv} (mW)	21.148	22.475	22.488	22.484

Après la première perturbation, i.e. les itérations 101 à 200, on voit que le P&O a convergé beaucoup plus rapidement que l'OEPS, récoltant ainsi plus d'énergie. Cependant, il a été démontré que l'OEPS reste à l'optimum d'une unité sans osciller, contrairement au P&O qui trouve l'optimum mais garde toujours une oscillation. Donc, pour une durée plus longue, l'OEPS aurait récolté plus d'énergie. On confirme d'ailleurs cette observation en comparant les P_{conv} des algorithmes. En effet, la puissance finale obtenue par l'OEPS est plus élevée que celle du P&O, donc une optimisation à plus long terme aurait récolté plus d'énergie, car P_{conv} représente en fait le gradient de l'énergie récoltée par rapport au temps (P_{conv} est en Watts, et $1\text{ W} = 1\text{ J/s}$). Les puissances finales obtenues par l'OEPS, l'OEPP, et l'OEPPC sont très semblables. En effet, tous ces algorithmes profitent de la précision de l'OEP, donc il est normal que ces algorithmes réussissent à trouver les optima de façon semblable et de façon plus précise que le P&O. Les algorithmes OEPP et OEPPC se distinguent toutefois par leur capacité de converger rapidement à l'optimum. L'OEPP a convergé plus rapidement que l'OEPPC, mais l'OEPPC a tout de même récolté une quantité d'énergie semblable à l'OEPP.

3.2.2.3 Optimisation d'unités non identiques en présence de perturbations

Dans ce cas d'étude, 10 PCM ayant des courbes statiques de puissance différentes sont utilisées. Les valeurs initiales de la concentration du substrat d'alimentation sont : $S_{0_{init}} = [700, 700, 650, 650, 600, 600, 675, 675, 625, 625]$ pendant 100 itérations. À la centième itération, une perturbation est appliquée sur la concentration S_0 de chaque PCM, où $S_0 = S_0 - 400$. Ainsi, les concentrations S_0 des unités deviennent $S_{0_2} = [300, 300, 250, 250, 200, 200, 275, 275, 225, 225]$, et ce pendant 100 itérations. Puis, une nouvelle perturbation ($S_0 = S_0 + 400$) ramène la concentration S_0 de chaque unité à sa valeur initiale $S_{0_{init}}$ pendant 100 itérations. Au total, les algorithmes optimisent les PCM pendant 300 itérations, ou 720 heures. Les courbes statiques de puissance des PCM sont montrées à la Figure 3.7.

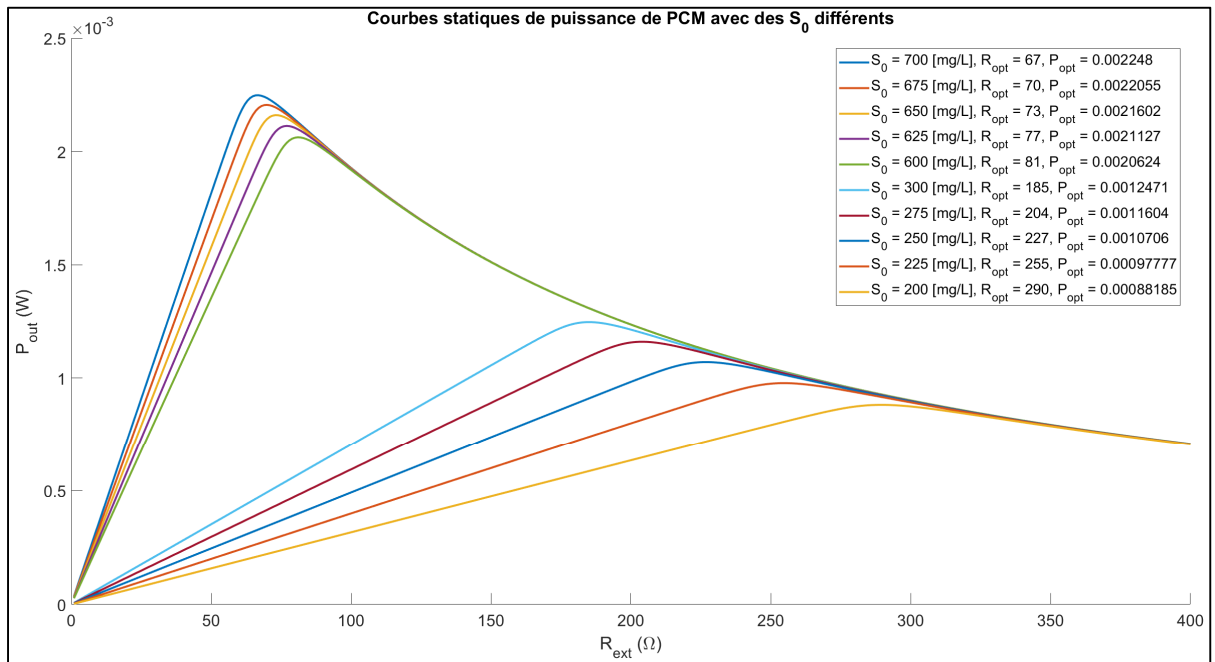


Figure 3.7 Courbes statiques de puissance de PCM à différents S_0

Normalement, comme dans le cas d'étude précédent, les performances des approches OEPS et P&O devraient être semblables, car ces algorithmes optimisent chaque PCM individuellement. Ce cas d'étude devrait cependant montrer la force de l'OEPPC par rapport à l'OEPP, et par rapport aux deux autres algorithmes, car l'OEPPC prend en compte les

différences des unités tout en gardant l'avantage de vitesse de l'OEPP, contrairement à l'OEPP qui considère que les unités sont toujours identiques.

La Figure 3.8 montre une simulation de la méthode P&O pour ce cas d'étude. On peut voir que la performance de l'algorithme est semblable au cas d'étude précédent. En effet, comme l'algorithme optimise chaque unité individuellement, les différences de performance ne seront dictées que par la position du R_{opt} des unités.

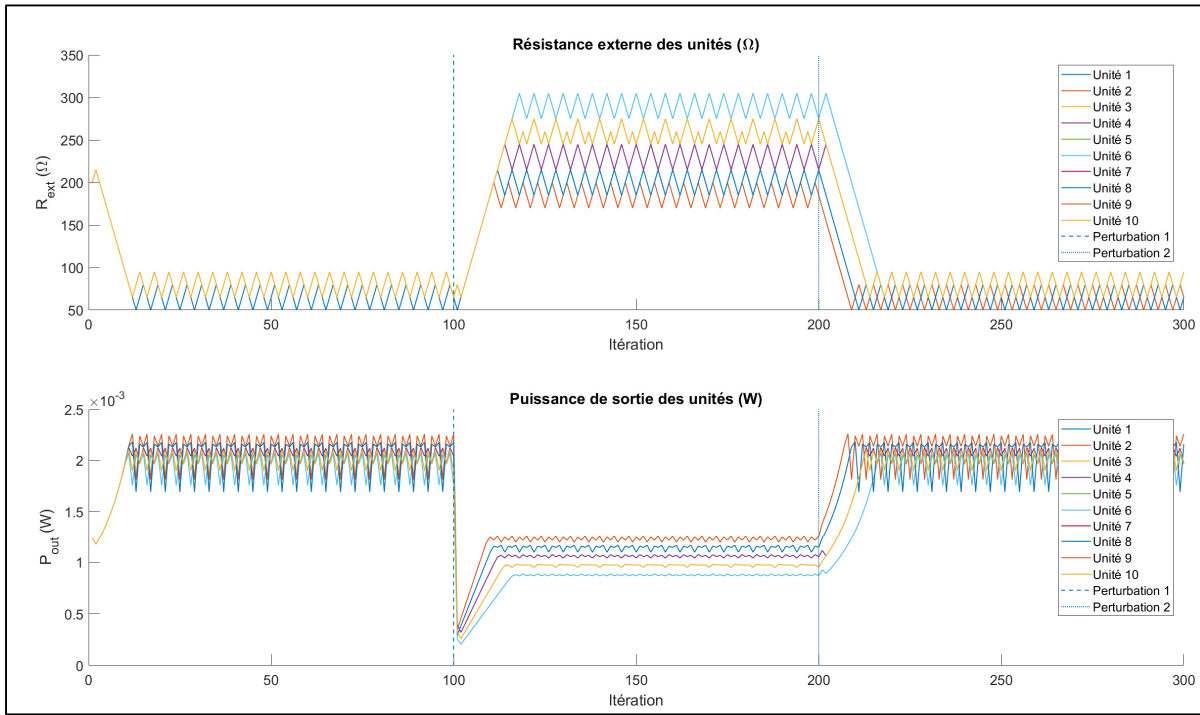


Figure 3.8 Optimisation par P&O de PCM non identiques en présence de perturbations

La Figure 3.9 montre une simulation de l'OEPS pour ce cas d'étude. Tout comme la méthode P&O, les performances de l'OEPS dans ce cas d'étude sont semblables à celles du cas d'étude précédent. La Figure 3.10 montre une simulation de l'OEPP pour ce cas d'étude. C'est ici que l'on peut voir que l'OEPP n'est pas idéale lorsque les différences des unités sont trop hautes car, bien que l'algorithme ait convergé très rapidement, certaines des unités ont convergé à une position qui n'est pas leur optimum, ce qui réduit la quantité d'énergie récoltée pendant l'optimisation. En effet, selon la nature de l'OEP, les particules tendent à

converger vers le *Gbest* de l'essaim. Or, pour le cas d'étude, le *Gbest* représente la position de la particule ayant la puissance optimale la plus haute. Ainsi, comme la position optimale de cette dernière n'est pas identique à celle des autres particules, les particules ayant une puissance optimale moindre convergent à un point qui n'est pas leur optimum.

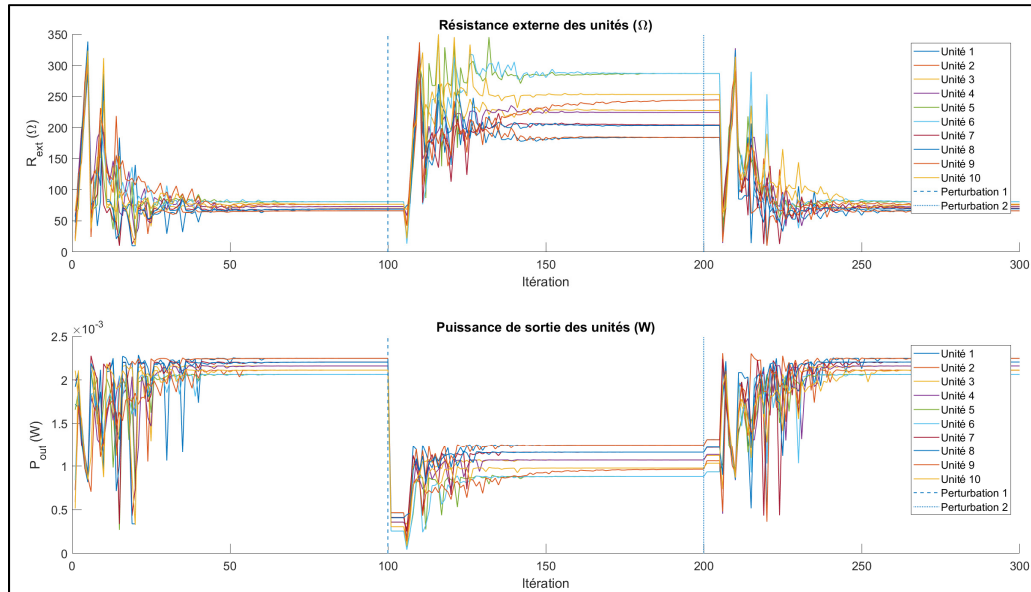


Figure 3.9 Optimisation par OEPS de PCM non identiques en présence de perturbations

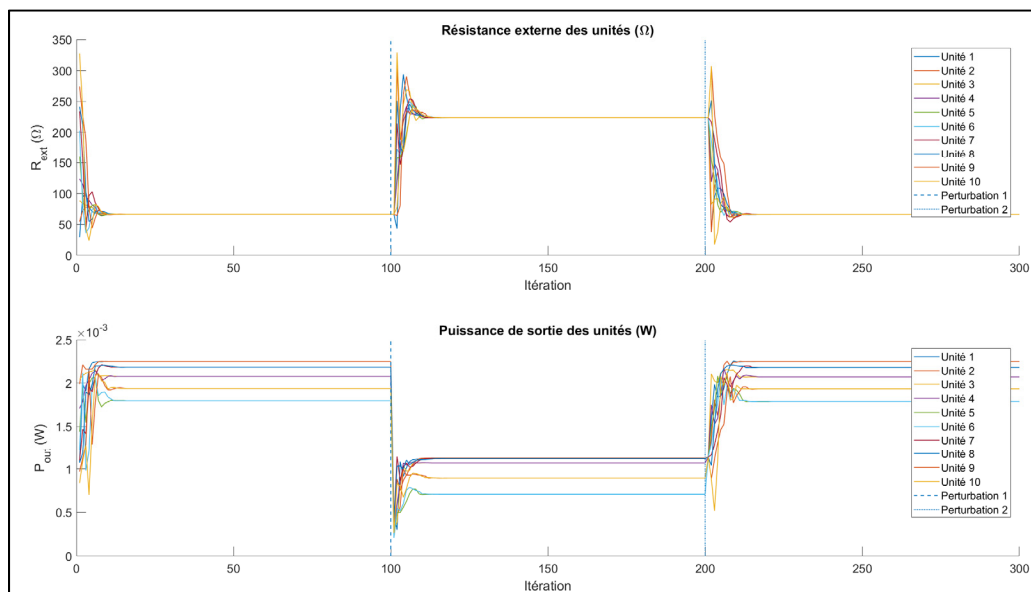


Figure 3.10 Optimisation par OEPP de PCM non identiques en présence de perturbations

La Figure 3.11 montre une simulation de l'OEPPC pour ce cas d'étude. On peut observer que l'algorithme converge très rapidement, mais qu'un ajustement est fait sur les unités qui ne sont pas à leur optimum. Par exemple, si l'on regarde l'unité 6 (en turquoise), après la première perturbation ($100 < \text{itération} < 201$), lorsque l'unité a convergé (itération ~ 125), l'unité est placée à son D_{opt} , qui a été réinitialisé lors de la perturbation et a été mis à jour pendant l'optimisation, puis une recherche par P&O est débutée, et D_{opt} est de nouveau réinitialisé. Lorsque l'optimisation par P&O appliqué sur l'unité 6 a convergé une première fois (itération ~ 132), son amplitude de perturbation Δ est diminuée de $\Delta_1 = 5 \Omega$ à $\Delta_2 = 1 \Omega$. Puis, lorsque l'unité 6 converge à nouveau (itération ~ 140), on place l'unité à son D_{opt} pour qu'elle y reste. On profite ainsi de l'avantage de vitesse de l'OEPP tout en affinant la position finale des unités avec l'approche P&O. On observe aussi sur la Figure 3.11 qu'après la première perturbation (itération 101), les unités ont convergé en groupe à deux points différents ($\sim 180 \Omega$ et $\sim 260 \Omega$). Effectivement, le classificateur a regroupé les unités en deux SA d'OEPP distincts, et le temps de convergence global a été amélioré. On voit ainsi l'avantage de l'ajout du classificateur à l'algorithme.

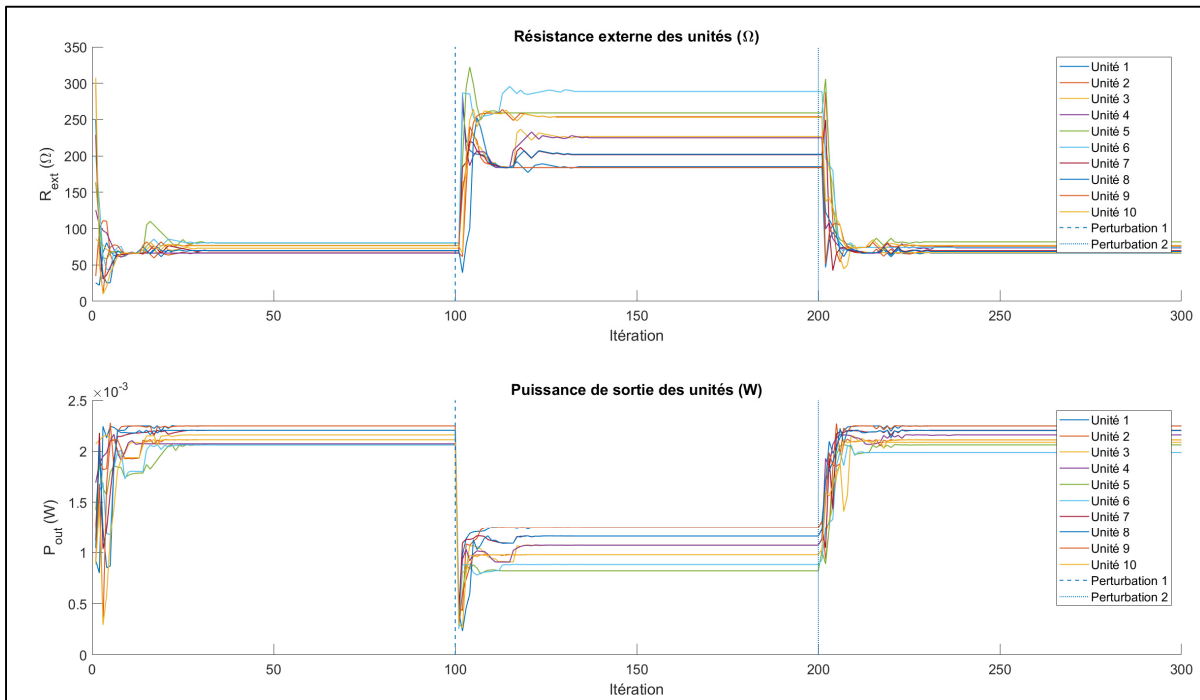


Figure 3.11 Optimisation par OEPPC de PCM non identiques en présence de perturbations

Le Tableau 3.6 compile les résultats des simulations pour ce cas d'étude. Comme au cas d'étude précédent, le tableau est divisé en 3 sections, soient : itérations 1 à 100 (optimisation initiale), itérations 101 à 200 (première perturbation), et itérations 201 à 300 (deuxième perturbation). Les meilleures performances pour chaque section ont été mises en gras et en italique. Les performances du P&O ne sont pas analysées pour la première section, car le temps de convergence plus long est dû à une valeur initiale de l'algorithme loin de l'optimum des unités.

Tableau 3.6 Résultats de simulation de PCM non identiques en présence de perturbations

Itérations	Critère	P&O	OEPS	OEPP	OEPPC
[1, 100] $S_0 = S_{0_{init}}$	E_{total} (kJ)	17.022	17.452	17.535	<i>18.204</i>
	T_{conv} (itérations)	11.4	62.03	<i>8.07</i>	24.12
	P_{conv} (mW)	20.413	<i>21.583</i>	20.650	21.580
[101, 200] $S_0 = S_{0_2}$	E_{total} (kJ)	8.625	8.497	7.879	<i>9.000</i>
	T_{conv} (itérations)	29.20	56.10	<i>9.66</i>	24.26
	P_{conv} (mW)	10.533	10.674	09.246	<i>10.676</i>
[201, 300] $S_0 = S_{0_{init}}$	E_{total} (kJ)	16.934	16.893	17.442	<i>18.152</i>
	T_{conv} (itérations)	12.20	67.37	<i>9.53</i>	26.17
	P_{conv} (mW)	20.410	21.553	20.655	<i>21.579</i>

Tout comme le cas d'étude précédent, les performances de l'approche P&O semblent légèrement meilleures que celles de l'OEPS. Cependant, comme il a déjà été mentionné, l'absence d'oscillations en régime permanent de l'OEPS résulterait en quantité d'énergie récoltée par l'OEPS à celle récoltée par la méthode P&O. Ceci est confirmé par une puissance moyenne en régime permanent P_{conv} supérieure pour l'OEPS que pour la méthode P&O.

On peut voir que l'OEPP est toujours l'algorithme qui converge le plus rapidement, mais il est aussi celui qui présente les pires performances au niveau de l'énergie récoltée et de la puissance finale des unités. En effet, on voit bien le désavantage d'ignorer les possibles différences entre les unités. Ainsi, l'OEPPC présente les meilleures performances au niveau du nombre de joules récoltées parmi tous les algorithmes. Ceci est dû au fait que l'OEPPC combine l'avantage de vitesse de l'OEPP, et la précision de l'OEPS et du P&O. L'OEPPC a aussi convergé beaucoup plus rapidement que l'OEPS, tout en obtenant une puissance finale semblable à celle de l'OEPS.

3.2.3 Synthèse des résultats

Cette section était dédiée à étudier et comparer les performances du P&O, de l'OEPS, de l'OEPP, et de l'OEPPC lorsqu'ils sont appliqués sur des PCM en simulation. Deux cas d'étude ont été présentés, soient : optimisation d'unités identiques avec perturbations, et optimisation d'unités non identiques avec perturbations. Les critères de performance utilisés sont ceux définis à la section 3.2.2.1.

Il a été démontré qu'à court terme, l'OEPS et le P&O ont des performances semblables. Cependant, grâce au fait que l'OEPS ne présente aucune oscillation en régime permanent, ce qui n'est pas le cas du P&O, l'OEPS récolterait plus d'énergie que le P&O à long terme si aucune perturbation ne survenait. Ce point a été prouvé avec le fait que l'OEPS menait toujours une puissance moyenne finale supérieure à celle obtenue avec l'approche P&O.

L'OEPP a aussi été démontrée comme étant très performante, mais seulement lorsque les unités optimisées présentent des courbes de puissance identiques ou très semblables. Lorsque des différences étaient présentes entre ces courbes, l'OEPP devenait beaucoup moins efficace. Aussi, il faut noter que l'OEPP performerait moins bien si des perturbations n'affectant que quelques unités du système survenaient car, dans ce cas, l'OEPP réinitialiserait la recherche pour toutes les unités, plutôt que de ne le faire que pour les unités perturbées. Ce contexte est d'ailleurs présenté plus en détails à l'ANNEXE III.

Lorsque les unités étaient différentes, l'OEPPC est l'algorithme qui, parmi tous les algorithmes étudiés, a permis de récolter le plus d'énergie, avec une puissance finale semblable à celle de l'OEPS. Le temps de convergence était similaire à celui du P&O, mais plus rapide que celui de l'OEPS. On peut expliquer ces résultats en analysant la structure même de l'algorithme. Tout d'abord, l'OEPP est utilisée pour converger le plus rapidement possible vers une zone qui est proche des optima des unités. On place ensuite les particules ayant convergé à leur D_{opt} respectif, puis on affine la position finale des unités avec le P&O. Après avoir convergé, les unités sont placées à leur D_{opt} respectif, mis à jour pendant l'optimisation par P&O, et y restent. Lorsque le système détecte une perturbation, les unités perturbées sont classifiées en fonction de leur optimum. Les unités ayant des optima semblables sont ainsi groupées et placées dans un SA d'OEPP. Il y a donc un SA d'OEPP par groupe créé par le classificateur. Cette étape est faite dans l'optique que l'OEPP converge assez bien lorsque les unités sont identiques ou semblables. Donc, en groupant des unités ayant un D_{opt} semblable, on augmente la capacité des SA d'OEPP à trouver une région proche des optima de toutes les unités du sous-algorithme.

Ainsi, en raison des meilleures performances de l'OEPPC, nous recommandons cet algorithme lorsqu'il est nécessaire d'optimiser un système comportant plusieurs unités pouvant présenter des différences dans leur courbes de puissance, et lorsque les unités pourraient être sujettes à des perturbations. Dans le cas où l'on est assuré que les unités sont identiques, l'OEPP est plus avantageuse. Ce cas de figure est toutefois très rare dans des applications réelles. S'il n'y a qu'une ou deux unités à optimiser, l'OEPS et l'approche P&O sont toutes deux recommandées, avec une préférence pour l'OEPS en raison de l'absence d'oscillations en régime permanent. Toutefois, si on ne peut tolérer les grandes variations initiales de R_{ext} , l'approche P&O est recommandée.

3.3 Évaluation expérimentale des algorithmes

Cette section présente les essais expérimentaux réalisés avec des PCM réelles. Les tests utilisent un système électronique d'acquisition et de commande développé dans le cadre de

ce mémoire, ainsi que des PCM développées à l'ÉTS au département de génie électrique. En raison de contraintes de temps, l'OEPPC n'a pas pu être appliquée sur les PCM, mais l'OEPS, l'OEPP, et le P&O ont pu être testés.

3.3.1 Procédure expérimentale

Cette section décrit d'abord les PCM utilisées, puis le système électronique d'acquisition et de contrôle utilisé est détaillé.

3.3.1.1 Piles à combustible microbiennes

Les PCM utilisées dans le cadre de cette étude ont été développées à l'ÉTS par le Groupe de Recherche en Électronique de Puissance et Commande Industrielle (GRÉPCI). Il est nécessaire de mentionner que la conception des PCM ne faisait pas partie de ce travail de recherche. Il s'agit de PCM solides, dans le sens où le substrat est une masse et non un liquide affluent, comme l'étaient les PCM utilisées en simulation à la section précédente.

Des feuilles de carbone avec catalyseur de manganèse ont été utilisées comme cathode (4 feuilles par PCM) et l'anode est composée de feutre de carbone traversé par un fil de titane de 24 cm (4 fils de titane et 4 morceaux de feutre de carbone par PCM). La structure de la pile est un boîtier rectangulaire imprimé en 3D, dont le modèle 3D est présenté à la Figure 3.12. La cathode a été fixée directement aux parois externes de la PCM afin d'être exposée à l'air, et l'anode aux parois internes, de façon à ce que le compartiment anodique soit anaérobique. Une membrane de polyester a été insérée entre les deux électrodes afin qu'il n'y ait aucun contact entre celles-ci. Le substrat a ensuite été inséré dans le compartiment anaérobique. Le substrat choisi était un mélange d'humus, de tourbe de sphaigne, de solution tampon, et d'eau sucrée. De la silicone a ensuite été appliqué aux joints de la pile pour en assurer l'étanchéité, et un contenant d'eau avec une ouverture a été placé sur le couvercle, de façon à assurer l'alimentation en eau de la pile en tout temps. Le montage final d'une PCM est présenté à la Figure 3.13.

En tout, trois PCM ont été construites de la sorte. Il a fallu attendre quelques semaines avant qu'elles soient assez stables pour pouvoir tester les algorithmes sur le système. Au moment de faire les tests, les PCM avaient une résistance interne d'approximativement $50 - 70 \Omega$, et une puissance de sortie de 2 à 3 mW. Il est spécifié que ces valeurs ont été prises avant les tests, car la dynamique interne des PCM varie continuellement, donc une mesure prise une semaine plus tard aurait mené à des valeurs de résistances internes différentes.

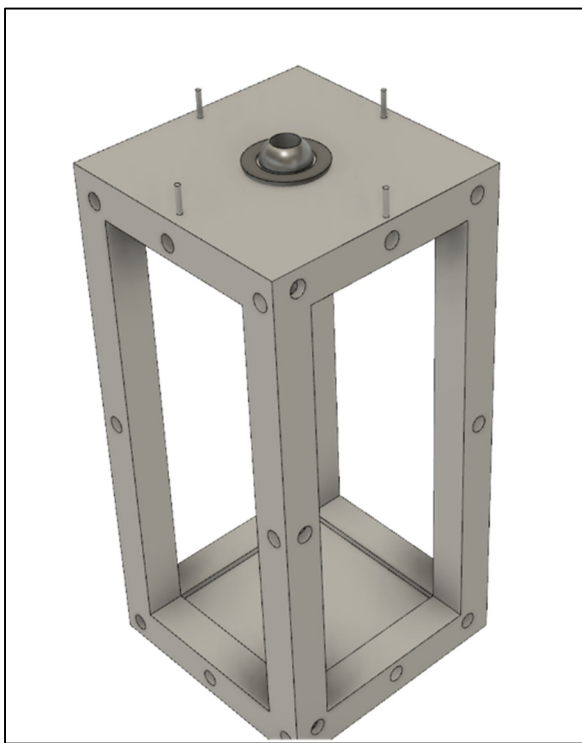


Figure 3.12 Modèle 3D du boîtier d'une PCM

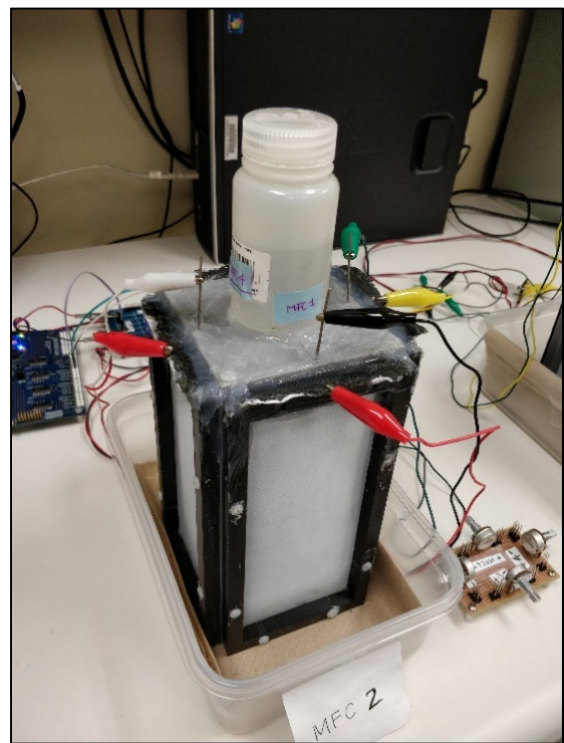


Figure 3.13 Montage d'une PCM

La Figure 3.14 montre la courbe statique de puissance des PCM. Ces courbes ont été obtenues en appliquant un potentiomètre aux bornes de chaque PCM et en le faisant varier d'une résistance de valeur R_{min} à R_{max} , puis de R_{max} à R_{min} par échelons de 3.92Ω et en attendant une heure entre chaque point d'opération. Le délai d'une heure a été choisi lors de tests préliminaires, où il a été déterminé que cette durée était suffisante pour laisser le temps aux PCM de se stabiliser après un changement de leur point d'opération. Pour les PCM 1 et 2, $R_{min} = 50 \Omega$ et $R_{max} = 101 \Omega$, et pour la PCM 3, $R_{min} = 61.76 \Omega$ et $R_{max} = 101 \Omega$. La

PCM 3 a un R_{min} différent car on ne voulait pas que sa puissance descende trop, de peur que la dynamique interne de la PCM soit impactée négativement. Cette limite a été descendue par la suite, égalisant ainsi les R_{min} de toutes les PCM. On peut observer que la puissance à un point d'opération n'est pas la même si l'on fait varier R_{ext} à la hausse que si l'on fait varier R_{ext} à la baisse. Il est donc difficile de déterminer avec précision la résistance interne d'une PCM. Ainsi, il est obligatoire d'utiliser une méthode de suivi du point de puissance maximal en temps réel.

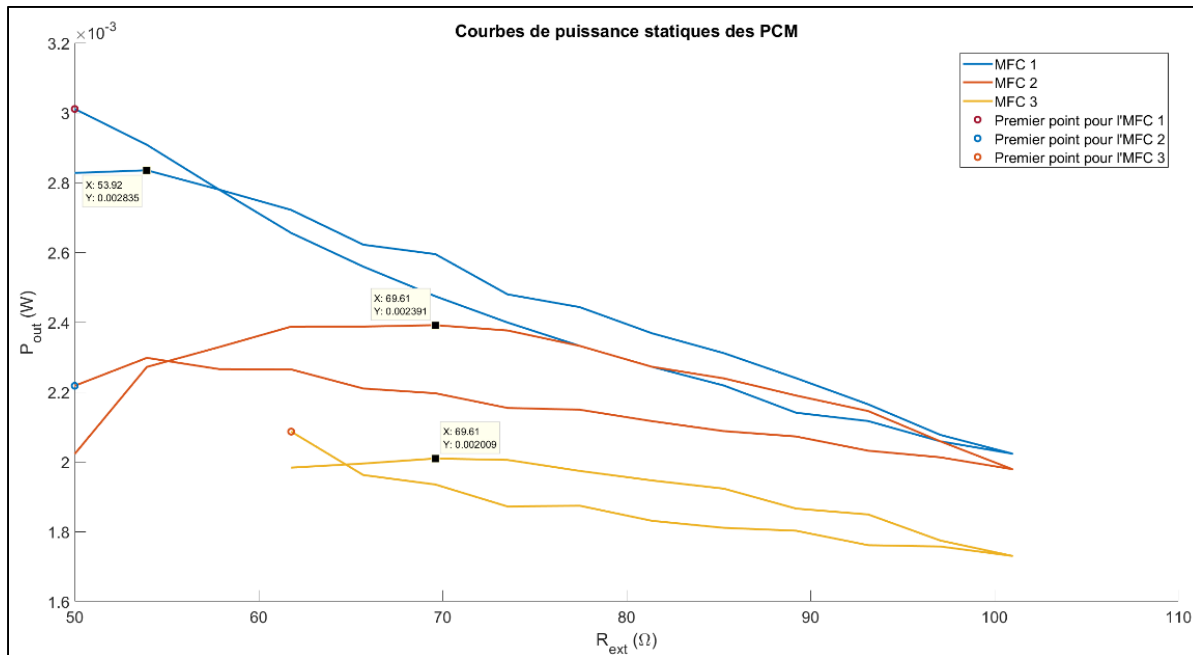


Figure 3.14 Courbes statiques de puissance des PCM

3.3.1.2 Système électronique d'acquisition et de commande

Un banc d'essai a initialement été conçu dans le cadre de ce mémoire pour tester des algorithmes sur un système PV. Ce banc d'essai a pu être adapté pour pouvoir tester expérimentalement des algorithmes d'MPPT sur des PCM. La Figure 3.15 est une photo du PCB développé. Les schémas électriques du PCB sont présentés à l'ANNEXE IV. Il s'agit d'un système embarqué composé d'un microcontrôleur PIC32MX795F512H (Microchip) et de quatre potentiomètres numériques AD8403ARZ1 (Analog Devices Inc). Un pilote de DEL

PCA9685 (NXP Semiconductors) est aussi présent sur le PCB, mais il n'est pas utilisé pour les essais expérimentaux sur les PCM (le CHAPITRE 4 parle plus en détails de l'utilité de ce pilote de DEL).

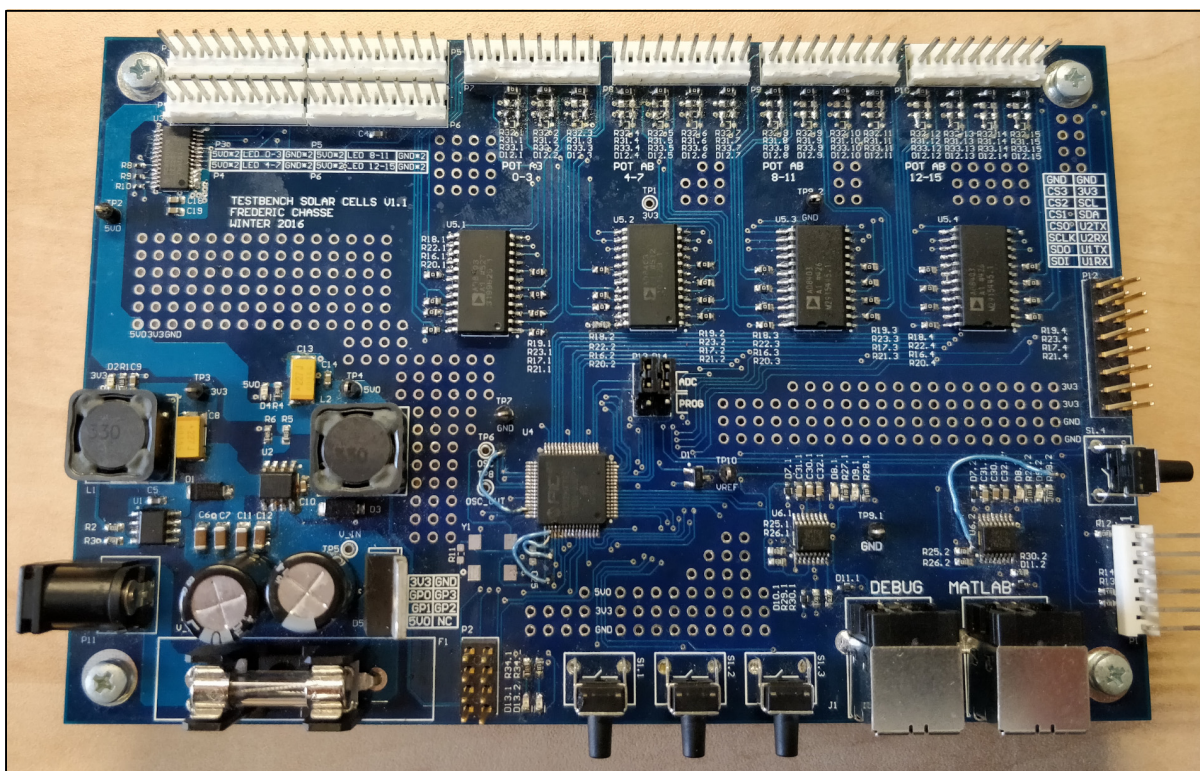


Figure 3.15 Carte électronique utilisée dans le banc d'essai

Le microcontrôleur intègre un convertisseur analogique/numérique à approximations successives (SAR ADC) ayant une résolution de 10 bits. Une tension de référence externe de 2.5 V lui est fournie (LM4040 de Diodes Incorporated), ce qui résulte en une résolution de 2.44 mV. La composante AD8403ARZ1 est un circuit intégrant 4 potentiomètres de 1 k Ω ayant 256 échelons (soit 3.9216 Ω par échelon), et chaque potentiomètre peut être contrôlé individuellement à l'aide de commandes envoyées par protocole SPI. Étant donné que le curseur de ces potentiomètres a une résistance de 50 Ω , on peut faire varier chaque potentiomètre de 50 Ω à 1050 Ω par incréments de 3.9216 Ω . En outre, un circuit de conversion UART à USB (FT230XS d'FTDI) se retrouve sur le PCB, ce qui permet d'établir

un canal de communication duplex intégral (« full duplex », en anglais) entre le microcontrôleur et un port sériel virtuel (« virtual COM port », en anglais) d'un ordinateur.

La Figure 3.16 montre le branchement d'une PCM au montage électronique. En résumé, l'électrode positive d'une PCM est connectée à un potentiomètre (« POT » dans la figure) et ce nœud électrique est aussi connecté à une entrée de l'ADC. L'électrode négative de la PCM est mise à la masse (« GND » dans la figure), de même que l'autre borne du potentiomètre. La valeur du potentiomètre est contrôlée de façon numérique par le microcontrôleur (« μC » dans la figure). L'ADC permet de lire la tension aux bornes de la PCM. En sachant la valeur de la charge connectée à la PCM, on peut calculer sa puissance de sortie avec l'équation (3.1), où V_{ADC} est la tension lue par l'ADC.

$$P_{PCM} = \frac{(V_{ADC})^2}{R_{ext}} \quad (3.1)$$

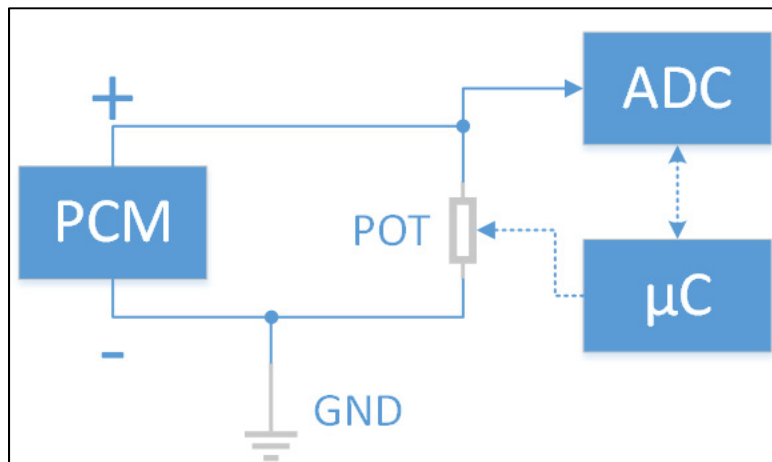


Figure 3.16 Représentation du branchement d'une PCM au circuit de contrôle et d'acquisition

Les différents algorithmes ont été codés en langage C sur le microcontrôleur et une interface a été faite dans MATLAB afin de récolter les données des expérimentations. Lors des tests, un échantillon pour chaque PCM était obtenu à chaque 1.2 s, et une moyenne de 10 échantillons par PCM était envoyée vers MATLAB. MATLAB recevait donc 3 mesures (moyennées) à chaque 12 secondes, ou une mesure par PCM. Une itération d'algorithme est

définie comme étant d'une durée d'une heure, donc une itération d'algorithme est faite après avoir reçu 300 mesures par PCM, donc 900 mesures au total. Il est à noter que seulement la 300^e mesure moyennée d'une PCM sera analysée par l'algorithme, car toutes les mesures précédentes représentent la dynamique des PCM. Or, le délai d'attente d'une heure est nécessaire afin d'obtenir la courbe de puissance statique des PCM. Toutes les mesures sont quand même envoyées vers MATLAB afin de pouvoir analyser la dynamique des PCM.

3.3.1.3 Génération de nombres pseudo-aléatoires

Étant donné que les algorithmes étudiés dans ce mémoire sont basés sur l'OEP (hormis le P&O), il était nécessaire de choisir une méthode de génération de nombres pseudo-aléatoires (NPA) offrant une distribution uniforme des nombres générés. Il existe plusieurs algorithmes de génération de NPA. Ces algorithmes requièrent tous une valeur d'initialisation (« seed » en anglais), et génèrent ensuite un nouveau NPA à chaque appel de l'algorithme selon une séquence qui sera différente pour chaque algorithme. La qualité de la séquence dictera la distribution des NPA générés. Or, pour une même valeur d'initialisation, un algorithme donné générera toujours la même séquence de nombres. Ainsi, il est important de pouvoir utiliser une valeur d'initialisation différente à chaque démarrage du système.

Il est assez difficile de générer une valeur d'initialisation aléatoire avec un microcontrôleur. Normalement, dans les systèmes embarqués, on peut générer un nombre aléatoire en ayant accès à une source de bruit que l'on mesurerait avec un ADC, ou en utilisant l'heure donnée par une horloge temps réel (« real-time clock » en anglais). Dans notre cas, nous avons accès à MATLAB pour générer cette valeur d'initialisation. En effet, la fonction « randi » de MATLAB permet de générer un entier sur 32 bits de façon pseudo-aléatoire avec la séquence du Mersenne Twister (Matsumoto et Nishimura, 1998). Ainsi, lors de l'initialisation du système, MATLAB envoie la valeur d'initialisation au microcontrôleur, qui initialise alors son algorithme de génération de NPA.

L'algorithme pseudo-aléatoire choisi pour cette étude expérimentale est de la famille des « *Permuted Congruential Generators* » (PCG) (O'Neill, 2014). L'algorithme utilise deux valeurs d'initialisation de 64 bits, et génère ensuite des NPA entiers de 32 bits distribués normalement. Cette méthode a été choisie car elle offre une bonne distribution, et aussi car l'auteur fournit une librairie codée en C sur son site web, ce qui a facilité le développement du code embarqué. Les valeurs d'initialisation de 64 bits sont en fait 2 entiers aléatoires de 32 bits générés avec la fonction « randi » de MATLAB et concaténés ensemble, puis envoyés au microcontrôleur par le port sériel.

3.3.1.4 Tests effectués

Ce banc d'essai a été utilisé pour caractériser la courbe de puissance statique des PCM et pour tester l'OEPS, l'OEPP, et la méthode P&O sur les PCM. En raison de contraintes de temps, il n'a pas été possible de tester l'OEPPC sur les PCM. Cependant, l'OEPPC a pu être testée sur un système composé de cellules photovoltaïques, et les résultats sont présentés au CHAPITRE 4. De plus, pour ces mêmes contraintes de temps et de matériel, il n'a pas été possible de créer un environnement perturbé pour les PCM. Ainsi, les cas d'études suivants sont faits :

1. Optimisation de trois PCM par OEPS;
2. Optimisation de trois PCM par OEPP;
3. Optimisation de trois PCM par P&O.

Le Tableau 3.7 donne les paramètres du P&O utilisés pour les tests et le Tableau 3.8 donne les paramètres de l'OEPS et de l'OEPP. On remarque dans le Tableau 3.8 que la marge des sentinelles est de $\pm 15\%$, ce qui est supérieur à ce qui était utilisé en simulation ($\pm 3\%$). La raison du choix de ce paramètre est que, pendant les tests préliminaires, il s'est avéré qu'un délai d'attente d'une heure pour passer d'un point d'opération à un autre était parfois insuffisant pour que la puissance d'une PCM se stabilise. Si une particule de l'OEPS restait à la même position pendant deux itérations et que la PCM n'avait pas encore atteint son régime

permanent, une différence de puissance dépassant la marge des sentinelles était lue, ce qui était détecté comme une perturbation par les particules, résultant en une réinitialisation de la recherche d'optimum. Pour remédier à ce problème, la marge de détection de perturbation des sentinelles a été augmentée à 15%. Ce problème est expliqué plus en détails à l'ANNEXE V.

Tableau 3.7 Paramètres de P&O pour les tests expérimentaux sur des PCM

Paramètre	Valeur	Paramètre	Valeur
Δ	7.8432 Ω	Position minimale	50 Ω
Position initiale	73.5294 Ω	Position maximale	202.9412 Ω

Tableau 3.8 Paramètres d'OEPS et d'OEPP pour les tests expérimentaux sur des PCM

Paramètre	Valeur	Paramètre	Valeur
c_1	0.9	Marge des sentinelles	$\pm 15\%$
c_2	1.7	Position minimale	50 Ω
ω	0.4	Position maximale	202.9412 Ω
Nombre de particules	3		

3.3.2 Résultats expérimentaux

La Figure 3.17 montre les résultats de l'optimisation de 3 PCM avec l'OEPS. La Figure 3.18 montre la même optimisation, mais n'affiche que la 300^e mesure des PCM pour chaque itération de l'algorithme dans le but de pouvoir observer l'évolution de l'algorithme sans la dynamique interne des PCM.

On observe tout d'abord à la Figure 3.17 de grandes variations de puissance lorsque la charge d'une PCM est changée. Par exemple, la puissance de la PCM 2 augmente grandement lorsque la charge est diminuée, mais cette puissance diminue rapidement par la suite. La Figure 3.19 montre uniquement le comportement de la PCM 2. On voit qu'à la 10788^e seconde, la résistance de la PCM est de 199.02 Ω , puis qu'à l'échantillon suivant (10800^e

seconde), la charge diminue à 50 Ω , ce qui résulte en une augmentation de la puissance pendant un bref moment. Cette variation peut être expliquée par la dynamique très lente des PCM. En effet, la tension qui était aux bornes de la PCM n'a pas eu le temps de changer lorsqu' R_{ext} a diminué. On peut calculer la tension aux bornes de la PCM en réorganisant l'équation (3.1) de la façon suivante :

$$P = \frac{V^2}{R} \rightarrow V = \sqrt{P * R}$$

$$V_{10788} = \sqrt{P_{10788} * R_{10788}} = \sqrt{0.00113845 \text{ W} * 199.02 \Omega} = 475.998 \text{ mV}$$

$$V_{10800} = \sqrt{P_{10800} * R_{10800}} = \sqrt{0.00399618 \text{ W} * 50.00 \Omega} = 447.000 \text{ mV}$$

$$V_{14388} = \sqrt{P_{14388} * R_{14388}} = \sqrt{0.00254898 \text{ W} * 50.00 \Omega} = 357.000 \text{ mV}$$

Ainsi, bien que 12 secondes se soient écoulées, la tension de la PCM à 10788 secondes et à 10800 secondes n'a pas beaucoup changé. On observe tout de même que la puissance décroît à un taux très rapide au début, puis commence à se stabiliser plus lentement.

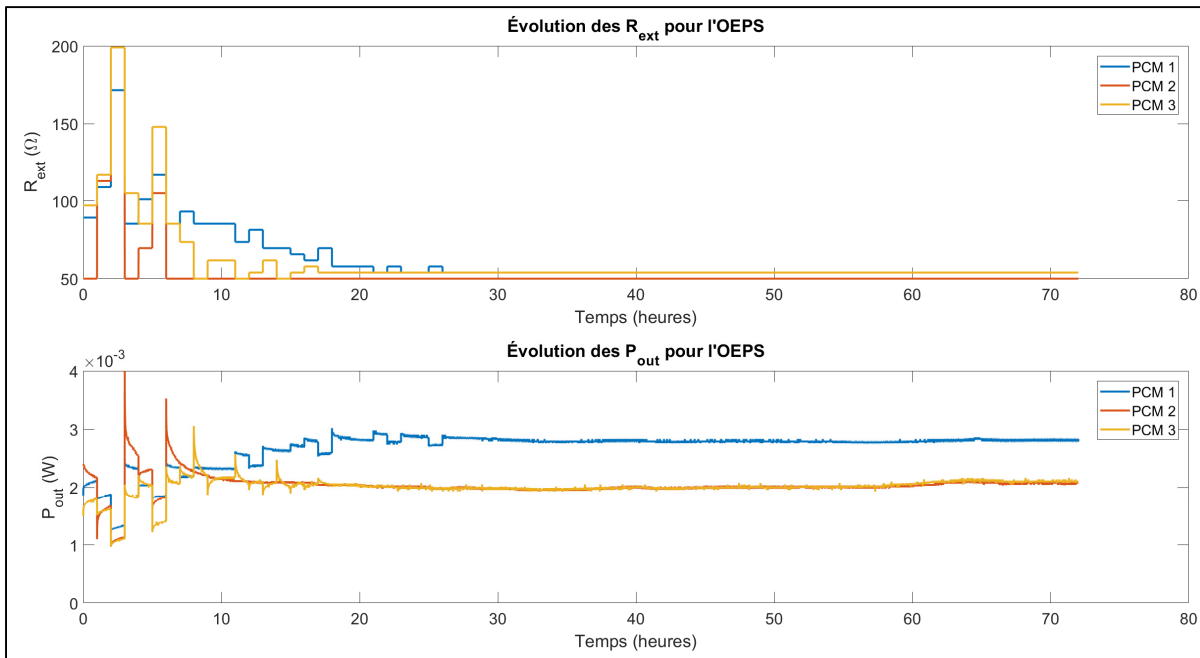


Figure 3.17 Données brutes de l'optimisation de 3 PCM par OEPS

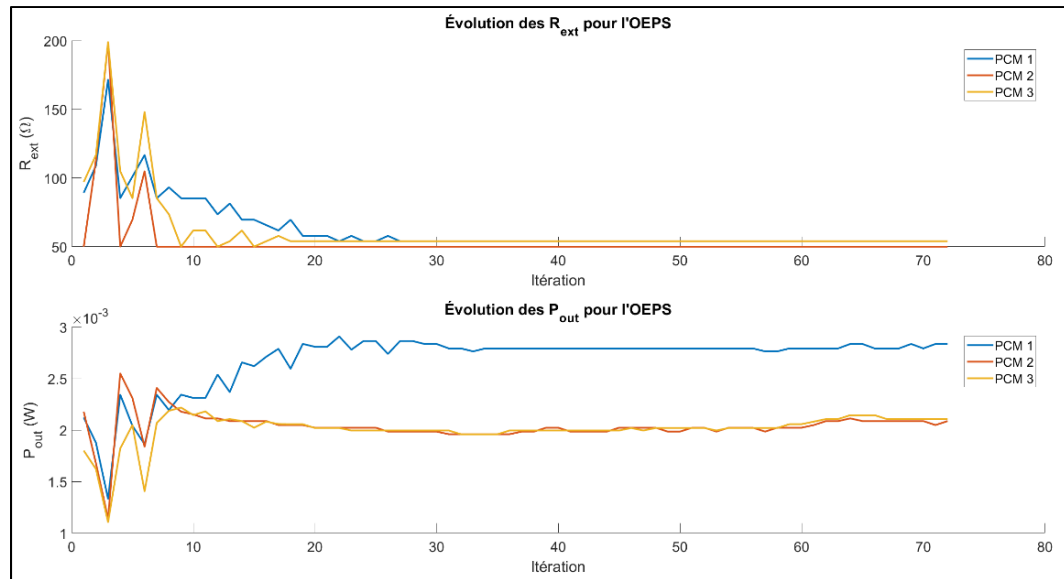


Figure 3.18 Données algorithmiques de l'optimisation de 3 PCM par OEPS

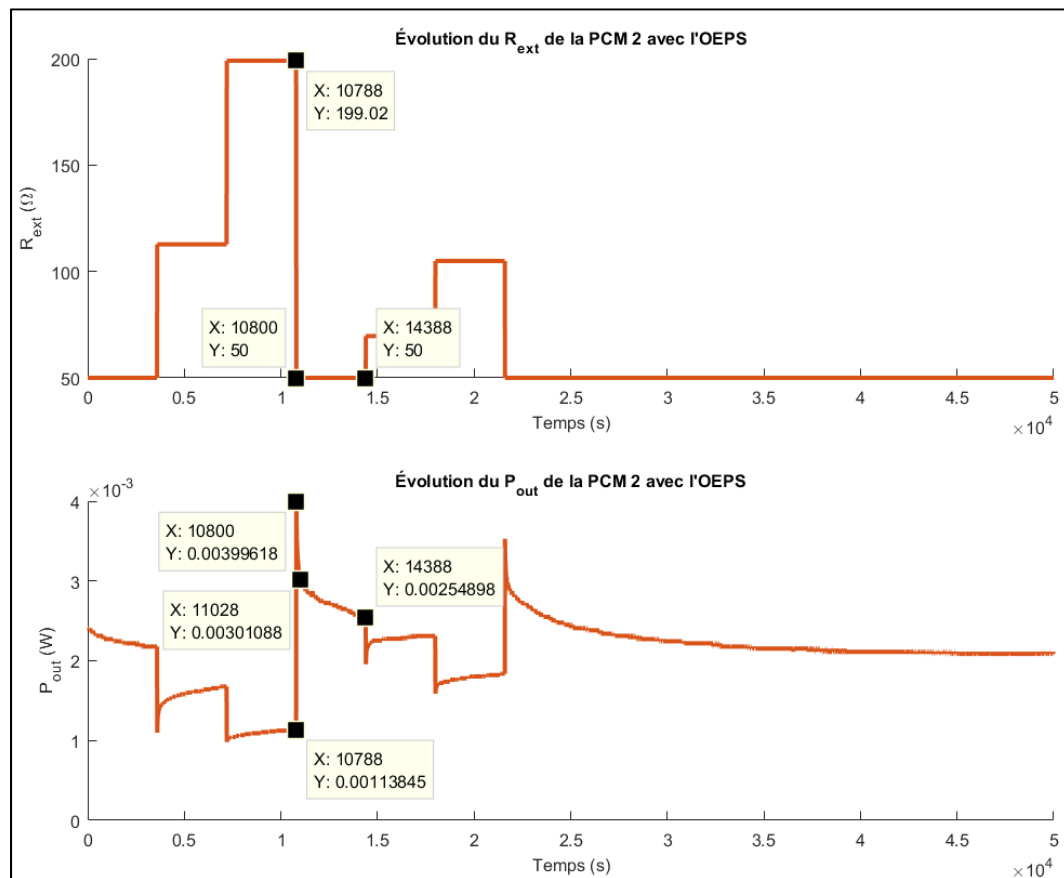


Figure 3.19 Analyse de la PCM 2 lorsqu'optimisée par OEPS

La Figure 3.20 montre l'évolution des tensions aux bornes des PCM lorsqu'elles sont optimisées avec l'OEPS. La courbe de tension lors d'un changement de résistance (R_{ext}) ressemble aux courbes de charge et de décharge d'un condensateur. Ainsi, on observe qu'à certaines itérations, les PCM n'étaient pas encore en régime permanent après une heure, tandis qu'elles étaient parfois stables après quelques minutes. On peut donc assumer que les mesures prises par l'algorithme lorsque les PCM n'étaient pas encore stables ne représentaient pas exactement leur courbe statique de puissance. L'OEPS a tout de même réussi à trouver le point maximal de puissance des PCM en à peu près 20 itérations, donc environ 20 heures.

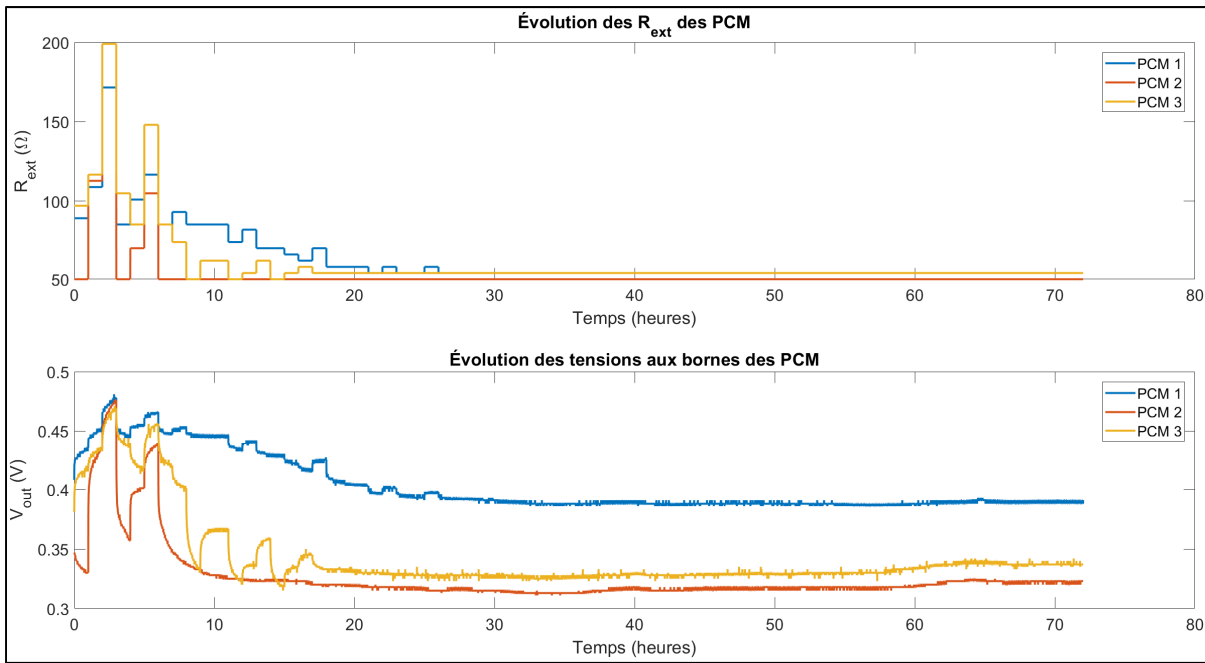


Figure 3.20 Évolution des niveaux de tension des 3 PCM optimisées par OEPS

En observant les résistances de la Figure 3.18, on voit que la PCM 1 a convergé en 27 itérations, la PCM 2 en 7 itérations, et la PCM 3 en 18 itérations. Les R_{ext} finaux des PCM 1, 2, et 3 étaient 53.92 Ω , 50 Ω , et 53.92 Ω , respectivement. La PCM 2 a convergé beaucoup plus rapidement, mais en analysant de plus près, on peut déduire que cette convergence rapide fait ressortir une faille de l'OEPS (et aussi de l'OEPP et de l'OEPPC). En effet, à la section 2.2.1, l'ajout d'une saturation à l'OEPS a été faite pour prendre en compte la

contrainte des limites inférieure et supérieure de la variable de contrôle. Ainsi, comme les meilleures valeurs de fitness obtenues étaient situées à la limite inférieure de la variable de contrôle (i.e. 50 Ω), les vitesses des particules calculées par l'algorithme ont rapidement diminué à zéro.

Le Tableau 3.9 représente l'évolution des particules de l'OEPS pour la PCM 2 pour les 4 premières itérations, où \vec{d} est un vecteur contenant les positions des particules 1, 2, et 3, \vec{f} est un vecteur contenant leur valeur de fitness en mW, \vec{Pbest} est un vecteur contenant les $Pbest$ des 3 particules, et \vec{v} un vecteur contenant leur vitesse. On observe que la particule 1 est restée à 50 Ω à chaque itération. Cela s'explique en regardant le calcul de la vitesse d'une particule (équation (1.9)). On voit que, si $d(i)$ est égal à $Pbest(i)$ et aussi à $Gbest(i)$ (ce qui était le cas de la particule 1 à la première itération), la vitesse de la particule dépendra du coefficient d'inertie de l'équation (1.9). Cependant, à la première itération, les vitesses sont assignées aléatoirement. Ainsi, si la vitesse de la particule 1 était négative (par exemple -10), le calcul de la prochaine position de la particule aurait été comme suit :

$$d(i + 1) = d(i) + v(i) = 50 - 10 = 40$$

Or, selon le pseudocode présenté à la section 2.2.1, toute valeur de position plus petite que la limite inférieure de la variable de contrôle (50 Ω dans notre cas) est fixée à celle-ci, et la vitesse est ajustée. Donc, pour ce cas spécifique, la vitesse serait fixée à zéro de la façon suivante :

$$v(i) = d(i + 1) - d(i) = 50 - 50 = 0$$

Ainsi, étant donné que 50 Ω reste le $Gbest$ de l'algorithme pour toute la durée de l'optimisation, la particule 1 ne bougera plus de cette position, et les deux autres particules convergeront aussi vers cette position. Ce problème aurait pu être évité si le potentiomètre utilisé pouvait descendre plus bas que 50 Ω , mais il n'a pas été possible de retravailler le banc d'essai. Il serait aussi possible d'éviter ce problème en empêchant l'initialisation de

l'algorithme de positionner une particule à une des deux limites. Il serait aussi faisable de forcer une vitesse différente lorsque la vitesse calculée pour une particule l'amène au-delà de l'une des limites. Néanmoins, la méthode la plus efficace serait probablement de s'assurer que les limites permettent à l'algorithme d'effectuer une recherche efficace autour des optima des unités. Il faut cependant noter que la valeur de 50 Ω s'est tout de même avérée être proche de l'optimum de la PCM 2, donc l'optimisation s'est par chance bien déroulée.

Tableau 3.9 Évolution des particules de l'OEPS sur la PCM 2

Variable	Itération 1	Itération 2	Itération 3	Itération 4
\vec{d}	[50 112.7 199]	[50 69.61 104.9]	[50, 50, 50]	[50, 50, 50]
\vec{J}	[2.18, 1.68, 1.14]	[2.55, 2.31, 1.84]	[2.41, 2.27, 2.18]	[2.15, 2.11, 2.11]
\overrightarrow{Pbest}	[50, 112.7, 199]	[50, 69.61, 104.9]	[50, 50, 50]	[50, 50, 50]
$Gbest$	50	50	50	50
\vec{v}	[0, -43.09, -94.1]	[0, -19.61, -54.9]	[0, 0, 0]	[0, 0, 0]

La Figure 3.21 et la Figure 3.22 montrent respectivement les données brutes et les données algorithmiques (i.e. les données recueillies à chaque itération de l'algorithme) de l'optimisation de 3 PCM avec l'OEPP. Comme en simulation, on voit que les particules de l'OEPP convergent beaucoup plus rapidement que celles de l'OEPS. Cependant, la limite inférieure a été atteinte et, à l'instar de l'OEPS, les particules ont rapidement cessé de bouger. Le point de convergence des particules est cependant proche de l'optimum des PCM, donc on observe quand même l'avantage de vitesse de l'OEPP.

On observe aussi une descente graduelle de la puissance de la PCM 1. En effet, nous avons découvert après les tests que la PCM 1 avait une fuite d'eau, ce qui faisait descendre sa puissance optimale. Normalement, ce changement aurait été détecté comme une perturbation par l'OEPP, mais vu que le paramètre de limite de détection de perturbation a été augmenté à 15% pour les essais expérimentaux, l'algorithme n'a pas détecté de variation assez grande pour justifier une nouvelle recherche.

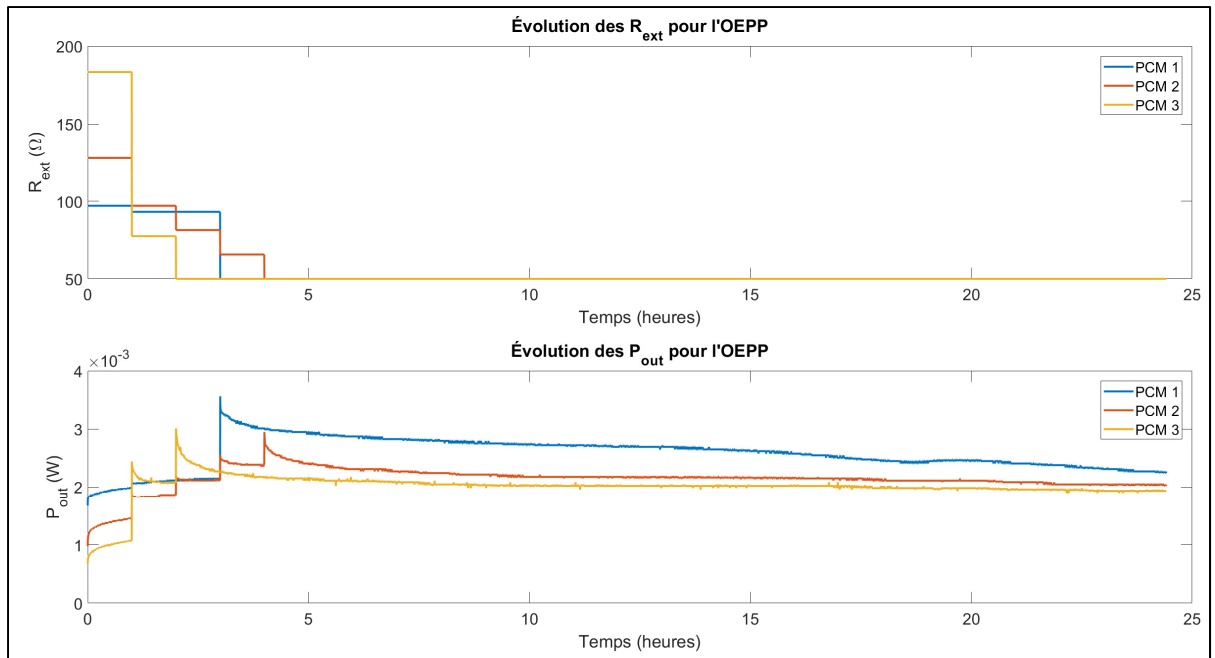


Figure 3.21 Données brutes de l'optimisation de 3 PCM par OEPP

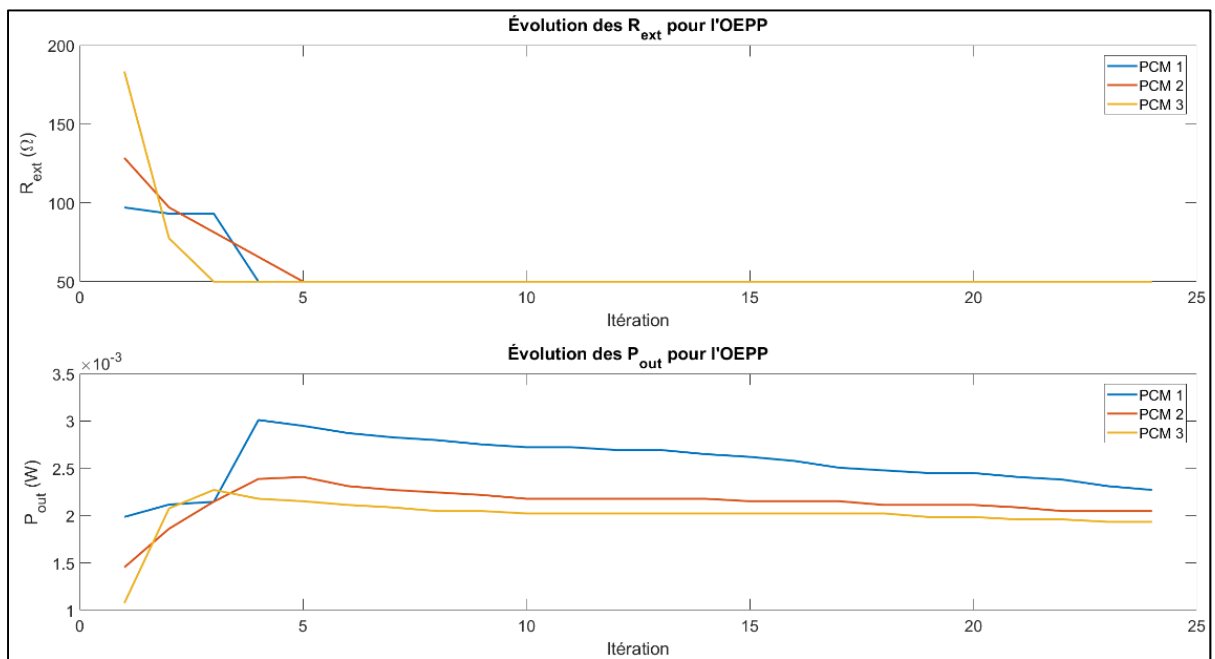


Figure 3.22 Données algorithmiques de l'optimisation de 3 PCM par OEPP

La Figure 3.23 montre les données brutes de l'optimisation de 3 PCM avec le P&O, et la Figure 3.24 montre les données algorithmiques de la même optimisation. On observe que le P&O trouve l'optimum des PCM et y reste, mais que l'oscillation autour de l'optimum entraîne d'assez grandes variations de puissance. Cependant, ces variations auraient été moins grandes si le Δ utilisé avait été plus petit. En effet, pour les tests, Δ était égal à deux fois la résolution des potentiomètres, c'est-à-dire $\Delta = 2 * 3.9216 \Omega = 7.8431 \Omega$. En raison de contraintes de temps, le test n'a pu être effectué avec $\Delta = 3.9216 \Omega$, mais on peut assumer que les variations de puissance auraient eu une amplitude moins grande.

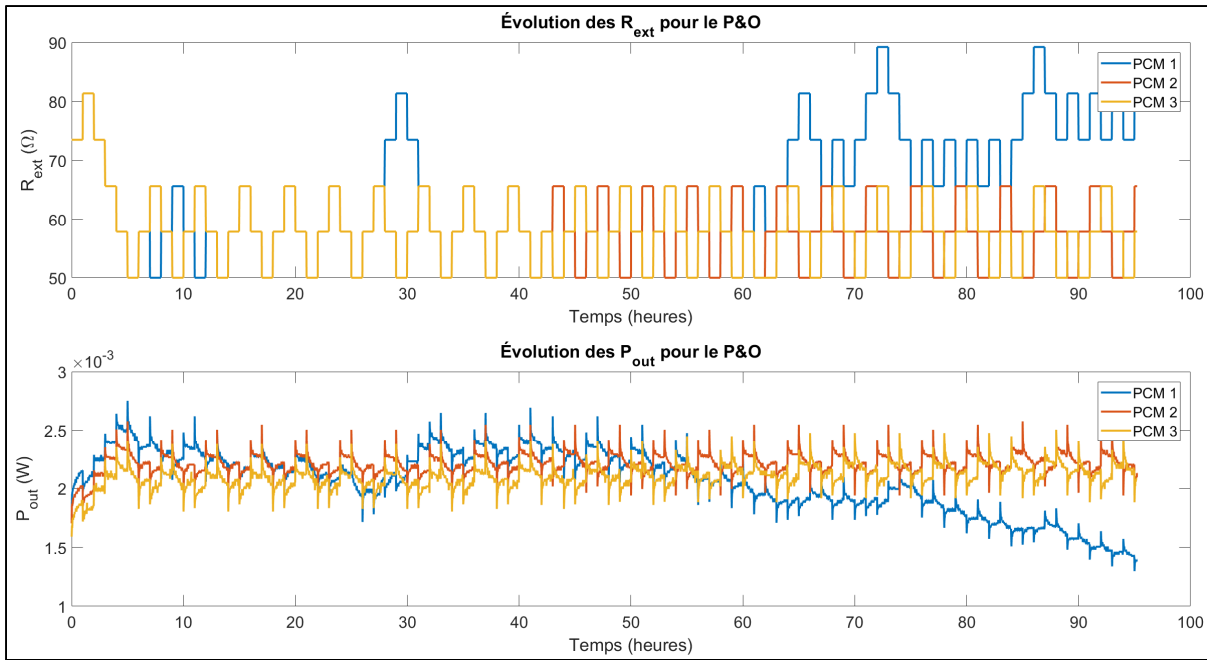


Figure 3.23 Données brutes de l'optimisation de 3 PCM par P&O

Ce test a été démarré peu après avoir terminé le test d'OEPP. Ainsi, la chute de puissance sur la PCM 1 observée pendant l'optimisation par OEPP est aussi présente pour l'optimisation par P&O. Cela permet aussi de voir que le P&O est capable de suivre un optimum qui varie lentement, ce qui a été le cas pour la PCM 1. Or, dans un algorithme basé sur l'OEP, une perturbation de la sorte aurait résulté en une nouvelle recherche, donc la vitesse de convergence aurait été plus lente que le P&O. Ainsi, l'approche P&O pourrait être plus apte à suivre un optimum qui varie très lentement. Cependant, dans le cas où la puissance de

sortie d'une unité est très stable, les oscillations autour de l'optimum de l'approche P&O causeraient une diminution de l'efficacité de l'algorithme.

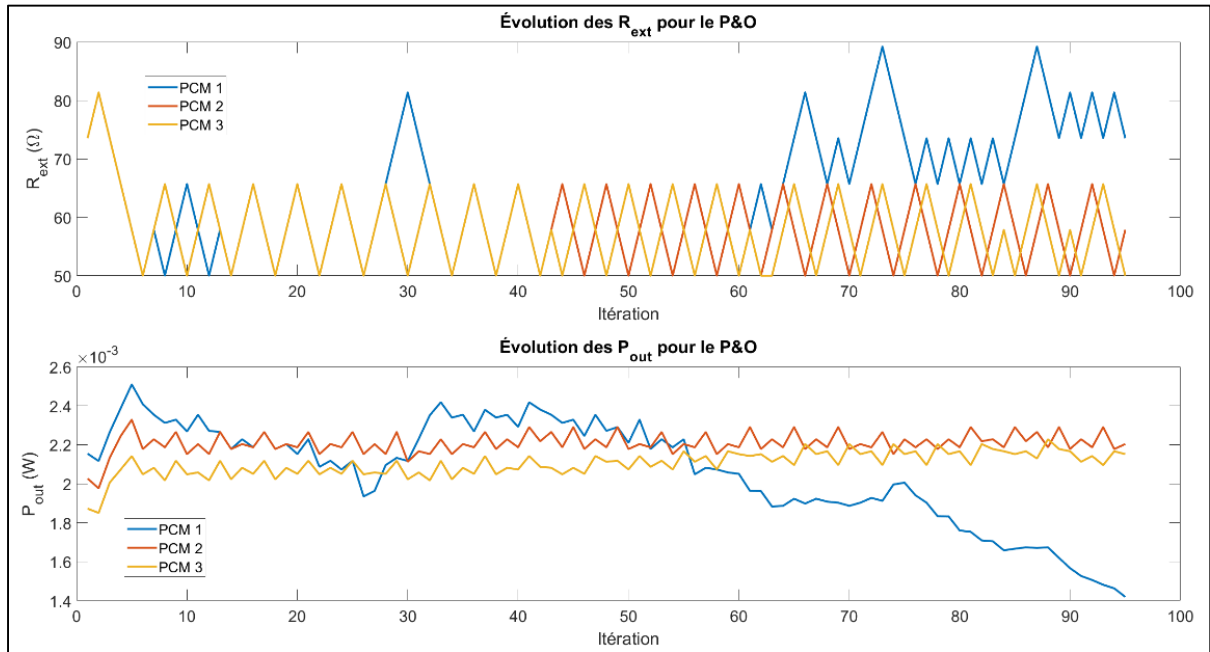


Figure 3.24 Données algorithmiques de l'optimisation de 3 PCM par P&O

3.3.3 Synthèse des résultats

Il est important de noter que les trois tests n'ont pas été faits sur la même base de temps. En effet, l'OEPS a d'abord été testée pendant 72 heures, l'OEPP a ensuite été testée pendant 24 heures, et le P&O a été testé par la suite pendant 95 heures. Étant donné la longue durée des tests, il a été décidé d'arrêter un test une fois la convergence atteinte. Ainsi, le test sur l'OEPP a duré beaucoup moins longtemps, et le test du P&O a duré plus longtemps car nous voulions observer le comportement de la PCM 1 dont la puissance diminuait. Afin d'avoir la même base de comparaison, les résultats expérimentaux sont évalués sur 24 heures, soit la durée du test le plus court, celui de l'OEPP.

Le Tableau 3.10 donne l'énergie (en joules) récoltée aux charges externes des PCM lorsque commandées par chacun des algorithmes (E_{total}), le Tableau 3.11 donne les temps de

convergence en heures de chaque algorithme (T_{conv}), et le Tableau 3.12 donne les puissances de sortie en mW obtenues par les algorithmes à partir du temps de convergence (P_{conv}). Les meilleures performances pour chaque colonne ont été mises en gras et en italique. Il est nécessaire de rappeler que la dynamique interne d'une PCM varie dans le temps, donc leur puissance optimale change aussi. Ainsi, la puissance optimale d'une PCM obtenue lors d'un test ne sera pas la même pour un autre. Les critères de comparaison E_{total} et P_{conv} sont donc analysés en tenant compte du fait que la puissance optimale d'une PCM a probablement changé d'un test à un autre.

Tableau 3.10 Énergie récoltée par les algorithmes lors des tests sur des PCM

Algorithme	E_{total} PCM 1 (joules)	E_{total} PCM 2 (joules)	E_{total} PCM 3 (joules)	$\sum E_{total}$ (joules)
OEPS	214.74	183.07	173.86	571.66
OEPP	225.76	188.09	176.36	590.20
P&O	197.52	192.76	180.05	570.32

Tableau 3.11 Temps de convergence des algorithmes lors des tests sur des PCM

Algorithme	T_{conv} PCM 1 (heures)	T_{conv} PCM 2 (heures)	T_{conv} PCM 3 (heures)
OEPS	23	5	17
OEPP	4	5	3
P&O	13	5	5

Tableau 3.12 Puissances finales obtenues par les algorithmes lors des tests sur des PCM

Algorithme	P_{conv} PCM 1 (mW)	P_{conv} PCM 2 (mW)	P_{conv} PCM 3 (mW)	$\sum P_{conv}$ (mW)
OEPS	2.82	2.10	2.03	6.95
OEPP	2.56	2.13	2.00	6.69
P&O	2.18	2.21	2.07	6.46

Il faut d'abord noter que la puissance de la PCM 1 a grandement diminué au fil du temps. En effet, en raison de la fuite d'eau de la PCM, la puissance obtenue pendant l'optimisation par OEPP était plus faible que pendant l'optimisation par OEPS, et encore plus faible lorsque la pile était optimisée par P&O. Cependant, on observe que l'OEPP est l'algorithme qui a récolté le plus d'énergie, et dont la vitesse de convergence est la plus rapide. En outre, bien que la puissance de la PCM 1 obtenue par l'OEPS était largement supérieure à celle récoltée par l'OEPP, l'OEPP a quand même récolté plus d'énergie que l'OEPS pour cette PCM. On peut ainsi affirmer que l'OEPP est meilleure que l'OEPS dans ce contexte grâce à sa vitesse de convergence beaucoup plus rapide, mais aussi surtout parce que les résistances internes des PCM étaient très semblables.

L'énergie totale extraite des PCM 2 et 3 à l'aide de la méthode P&O est cependant supérieure à celle récoltée avec les deux autres algorithmes. Nous pourrions *a priori* attribuer cette meilleure performance au fait que la puissance optimale de ces PCM semble avoir augmentée au fil du temps. En effet, il a été démontré dans la littérature qu'une PCM ayant sa charge externe équivalente à sa résistance interne voit sa puissance optimale augmenter (Pinto et al., 2011). Cela est confirmé au Tableau 3.12 pour la PCM 2, car sa puissance optimale a augmenté à chaque test. La puissance optimale de la PCM 3 est cependant restée très stable pendant tous les tests. Ainsi, pour la PCM 3, le P&O a récolté plus d'énergie que l'OEPS grâce à la convergence plus rapide de l'algorithme pour ce test. Cette dernière observation s'explique par le fait que la condition initiale du P&O était très proche de l'optimum réel des unités, ce qui a réduit le temps de recherche de l'algorithme, tandis que l'OEPS fait une recherche dans tout l'hyperespace avant de converger.

Un problème qui n'avait pas été considéré pendant les simulations est apparu dans les expérimentations. En effet, la fuite d'eau subie par la PCM 1 nous a démontré qu'il est possible que la dynamique interne d'une PCM varie très lentement, et donc que son optimum varie aussi très lentement. Or, les algorithmes ont tous été étudiés en simulations dans un contexte où de grandes perturbations ont été appliquées sur les PCM, mais le cas où des perturbations lentes et graduelles surviennent n'a pas été considéré. Cependant, on peut

facilement déduire que les algorithmes basés sur l'OEP seraient moins efficaces que le P&O dans ce contexte. Effectivement, les algorithmes présentés au CHAPITRE 2 réagissent tous de la même façon lorsqu'ils détectent une perturbation, i.e. les particules sont réinitialisées aléatoirement dans l'hyperespace, et une recherche de l'optimum recommence. Cette nouvelle recherche n'aurait toutefois pas été nécessaire car, bien que la position de l'optimum ait changé, elle n'aurait pas changé énormément. Ainsi, par déduction, on peut affirmer que l'approche P&O serait plus apte à suivre un optimum qui varie lentement que les algorithmes basés sur l'OEP.

En revanche, il serait possible d'adapter l'OEPPC pour ce cas spécifique. En effet, l'une des techniques utilisées dans les MPPT pour suivre un optimum consiste à introduire de la diversité pendant la recherche, même si aucune perturbation n'a été détectée (Nguyen, Yang et Branke, 2012). Ainsi, nous proposons d'ajouter une diversité à la partie P&O de l'OEPPC. Une implémentation très simple serait de compter le nombre d'itérations où une unité est restée fixée à D_{opt} après la recherche par P&O, puis lorsqu'un certain nombre d'itérations a été atteint, la recherche par P&O serait redémarrée sur l'unité. La Figure 3.25 donne un exemple de l'optimisation de 10 PCM non identiques par OEPPC avec ajout de diversité (OEPPCD). Dans ces résultats de simulation, les PCM ont subi une perturbation de $S_0 = S_0 - 0.25 \text{ mg/L}$ pendant 100 itérations à partir de l'itération 50.

On voit que l'algorithme réussit à suivre l'optimum des PCM qui varie lentement. D'ailleurs, une comparaison en simulation des performances de l'OEPPCD et du P&O est faite à l'ANNEXE VI, et l'OEPPCD performe de façon légèrement supérieure au P&O pour des perturbations lentes. Donc, dans le cas des PCM, où la dynamique très lente des piles peut faire varier l'optimum de façon lente, l'ajout de diversité à l'OEPPC est bénéfique. Bien entendu, dans le cas où aucune perturbation ne survient, l'OEPPCD perd de son efficacité par rapport à l'OEPPC en raison des oscillations rajoutées par le P&O. L'ANNEXE VI démontre toutefois que l'OEPPCD performe mieux que l'approche P&O dans un contexte où aucune perturbation ne survient, justifiant ainsi l'ajout de diversité à l'algorithme.

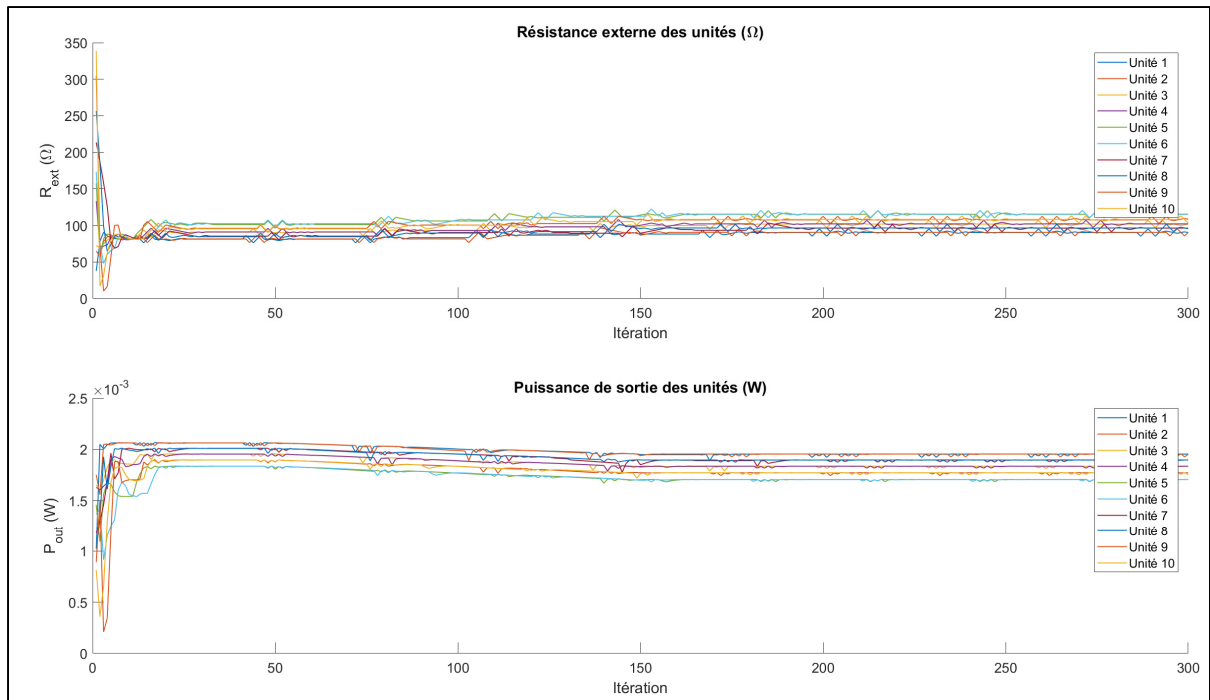


Figure 3.25 Exemple d'optimisation par OEPPC avec ajout de diversité de 10 PCM non identiques avec perturbations lentes

3.4 Conclusion

Cette section a présenté les performances des algorithmes OEPS, OEPP, OEPPC, et P&O appliqués sur un système composé de plusieurs PCM. La section 3.2 présentait les résultats des simulations des 4 algorithmes sur un système composé de 10 PCM, tandis que la section 3.3 comparait expérimentalement les performances des algorithmes OEPS, OEPP, et P&O appliqués sur 3 PCM développées dans les laboratoires du GRÉPCI à l'ÉTS.

Les résultats de simulation ont démontré que l'OEPS et la méthode P&O avaient des performances assez semblables à court-terme. Toutefois, à plus long terme, l'OEPS récolterait plus d'énergie que la méthode P&O en raison des oscillations induites par cette dernière approche autour de l'optimum. Dans le cas où les piles sont identiques, l'OEPP et l'OEPPC ont présenté des performances meilleures que l'OEPS et que l'approche P&O grâce à leur vitesse de convergence très rapide.

Cependant, lorsque des différences entre les courbes de puissance des PCM ont été introduites, les performances de l'OEPP ont grandement diminué, et l'OEPPC a été l'algorithme qui performait le mieux. En outre, si une perturbation survenait sur seulement quelques unités du système, l'OEPP redémarrerait la recherche pour toutes les unités, alors qu'il n'aurait été nécessaire de le faire que pour les unités perturbées. Ainsi, l'OEPP ne serait pratique que lorsque les unités d'un système sont identiques et qu'il est garanti qu'elles subiront les mêmes perturbations aux mêmes instants. Or, ces conditions sont irréalistes dans un contexte expérimental. Donc, l'OEPPC est l'algorithme que nous favorisons pour l'optimisation d'un système composé d'au moins 3 unités.

Les résultats expérimentaux ont démontré que l'OEPP performe mieux que l'OEPS en raison de sa vitesse de convergence beaucoup plus rapide. Cependant, il est vrai que les PCM présentaient des résistances internes semblables, donc dans le cas où ces résistances internes auraient été très différentes, on peut assumer que l'OEPS aurait mieux performé. Un problème concernant la limite inférieure de la variable de commande est ressorti pour l'OEPS et pour l'OEPP. Ce problème résulte en un arrêt de la recherche de l'optimum si certaines conditions sont remplies, e.g. la limite inférieure est proche de l'optimum des unités. Cependant, ce problème aurait pu être évité si les potentiomètres utilisés avaient permis d'atteindre une valeur inférieure à 50 Ω .

En raison de ce problème de limite inférieure, le P&O a mieux performé que les deux autres algorithmes car le P&O a pu trouver l'optimum des PCM de façon plus précise, tandis que l'OEPP et l'OEPS sont restés fixés à 50 Ω . En outre, un problème qui n'avait pas été considéré en simulations est apparu en expérimentations. La PCM 1 a commencé à perdre de l'eau pendant l'optimisation par OEPP, ce qui a fait diminuer sa puissance optimale lentement, mais cette diminution de puissance est vraiment devenue apparente pendant l'optimisation par P&O. L'observation qui est ressortie de ce changement très lent de la dynamique interne de la PCM est que le P&O performe mieux que les algorithmes basés sur l'OEP dans ce contexte. En effet, ces derniers redémarrent une recherche complète de l'optimum lorsqu'une perturbation est détectée, ce qui résulte en une grande variation de la

puissance aux bornes des unités optimisées. Ainsi, pour un optimum qui n'a pas beaucoup varié, ces algorithmes perdraient beaucoup d'efficacité. Le P&O, quant à lui, réussissait bien à suivre un optimum qui bouge lentement. Donc, dans un contexte où l'optimum d'une unité varie lentement, l'approche P&O serait plus adaptée pour suivre cet optimum en temps réel. Il reste cependant vrai que, si l'optimum est stable, l'approche P&O perd de son efficacité en raison des oscillations autour de l'optimum.

Cependant, il a été suggéré qu'il serait possible de rajouter une diversité à l'OEPPC afin de compenser un optimum qui varie lentement. De cette façon, l'algorithme garde l'avantage de vitesse de l'OEPP et la précision de l'OEPPC, mais acquiert aussi l'avantage du P&O lorsque l'optimum varie lentement. Une étude de cet algorithme est donnée à l'ANNEXE VI, et les performances de l'OEPPCD ont été jugées légèrement supérieures à celles du P&O lorsque les unités optimisées subissent des perturbations lentes. De plus, si les unités ne subissent pas de perturbations, ou si les perturbations sont très grandes, l'OEPPCD devient meilleure que le P&O. Ainsi, bien que nous n'ayons pas pu tester l'OEPPC expérimentalement, nous pensons que l'ajout de diversité à l'algorithme le rendrait apte à optimiser un système composé de plusieurs PCM.

Il est vrai que l'OEPPC et l'OEPPCD sont plus complexes que l'approche P&O au niveau de leur implémentation dans un microcontrôleur. Cependant, il a été calculé en simulation que l'OEPPC offre un gain en énergie récoltée d'environ 7% sur la durée des tests par rapport à l'approche P&O, et un gain en puissance finale de 5%. Donc, bien que les algorithmes prennent plus de temps à implémenter, le gain en énergie récoltée justifie l'effort supplémentaire de programmation, surtout dans le cas des PCM, dont la densité de puissance de sortie est très faible.

Comme il n'a pas été possible d'étudier expérimentalement les performances de l'OEPPC dans ce chapitre, le CHAPITRE 4 fait une étude expérimentale sur un système composé de plusieurs cellules photovoltaïques qui comparera les performances du P&O, de l'OEPS, et de l'OEPPC dans différentes conditions opératoires. Ce chapitre permet de comparer les

performances de l'OEPPC à celles d'algorithmes éprouvés sur les systèmes PV, et démontre que, bien que l'OEPPC ait été développée dans le but d'optimiser des PCM, elle s'applique aussi à d'autres types de systèmes dynamiques.

CHAPITRE 4

OPTIMISATION EN TEMPS RÉEL DE LA PUISSANCE DE SORTIE DE CELLULES PHOTOVOLTAÏQUES

4.1 Introduction

Ce chapitre présente une comparaison des performances des algorithmes P&O, OEPS, et OEPPC lorsqu'ils sont appliqués sur un système composé de cellules photovoltaïques. Le P&O et les algorithmes basés sur l'OEP ont déjà été testés sur les systèmes PV, et ont fait l'objet de plusieurs articles scientifiques. Le but de ce chapitre n'est donc pas d'étudier l'applicabilité de l'OEP ou du P&O à un système PV, mais plutôt de comparer les performances de l'OEPPC lorsqu'appliquée sur un système PV à celles d'algorithmes déjà éprouvés sur le même système. En outre, il n'a pas été possible de tester l'OEPPC sur des PCM, donc ce chapitre permet aussi de tester l'algorithme sur un système réel et d'étudier ses performances dans un tel contexte.

L'OEPP n'est pas étudiée dans ce chapitre car il a déjà été établi au CHAPITRE 3 que l'OEPP mène à des performances supérieures à l'OEPPC seulement lorsque les unités optimisées présentent des courbes de puissance statiques identiques. Lorsque les unités présentent des différences trop grandes, il a été démontré que l'OEPPC performe beaucoup mieux que l'OEPP. Or, les cellules PV utilisées dans ce chapitre présentent toutes des courbes de puissance différentes. Ainsi, il n'a pas été jugé utile de tester l'OEPP dans ce chapitre.

Deux cas d'étude ont été définis pour permettre d'établir la supériorité de l'OEPPC par rapport au P&O et à l'OEPS, soient : optimisation sans perturbations, et optimisation avec perturbations. La section 4.2 explique la procédure expérimentale utilisée pour les tests, et la section 4.3 présente les résultats expérimentaux.

4.2 Procédure expérimentale

4.2.1 Système électronique d'acquisition et de commande

Le banc d'essai présenté à la section 3.3.1.2 a été utilisé pour les tests sur des cellules PV. Il avait été mentionné que le PCB de contrôle et d'acquisition intègre un contrôleur de DEL, mais ce contrôleur n'était pas utilisé pour les tests sur les PCM. Or, ce circuit de DEL est nécessaire pour les tests sur un système photovoltaïque afin de pouvoir contrôler l'intensité lumineuse reçue par les cellules PV.

Le banc d'essai a initialement été conçu dans le but de tester les algorithmes sur des cellules photovoltaïques. C'est d'ailleurs ce banc d'essai qui a été utilisé dans notre article de conférence (Chassé et Woodward, 2016). Afin de pouvoir tester un système composé de plusieurs cellules PV, il était nécessaire d'utiliser un microcontrôleur ayant plusieurs entrées analogiques. Ainsi, le microcontrôleur choisi a été le PIC32MX795F512H de Microchip, car il compte en tout 16 entrées analogiques. Une des entrées est réservée comme tension de référence du convertisseur, ce qui résulte en une possibilité de 15 cellules PV au total.

Afin de pouvoir contrôler individuellement l'intensité lumineuse reçue par chaque cellule PV, le contrôleur à DEL PCA9685 (NXP Semiconductors) a été intégré au PCB de contrôle. Ce circuit intégré permet de contrôler individuellement l'intensité lumineuse de 16 DEL par PWM. Le microcontrôleur communique avec le PCA9685 par protocole I²C.

En outre, afin d'isoler le niveau de luminosité que chaque cellule PV reçoit, un montage mécanique a été élaboré. À cet escient, deux PCB supplémentaires ont été conçus, soient : un PCB de DEL, et un PCB de cellules PV. Le PCB de cellules PV est placé au fond d'un boîtier, et le PCB de DEL vient se placer sur le dessus du boîtier, de façon à ce que chaque cellule PV soit illuminée par 3 DEL. De plus, une pièce mécanique séparant les cellules PV est insérée dans le boîtier, permettant ainsi d'isoler l'intensité lumineuse reçue par chaque cellule. La Figure 4.1 donne le modèle 3D du boîtier et du mur de séparation, la Figure 4.2 montre l'assemblage 3D du PCB de cellules PV, du boîtier, et du mur de séparation, la

Figure 4.3 montre le PCB de cellules PV encapsulé dans le boîtier, la Figure 4.4 montre le dessus du PCB de DEL et la Figure 4.5 montre le dessous, et la Figure 4.6 montre le montage complet du banc d'essai.

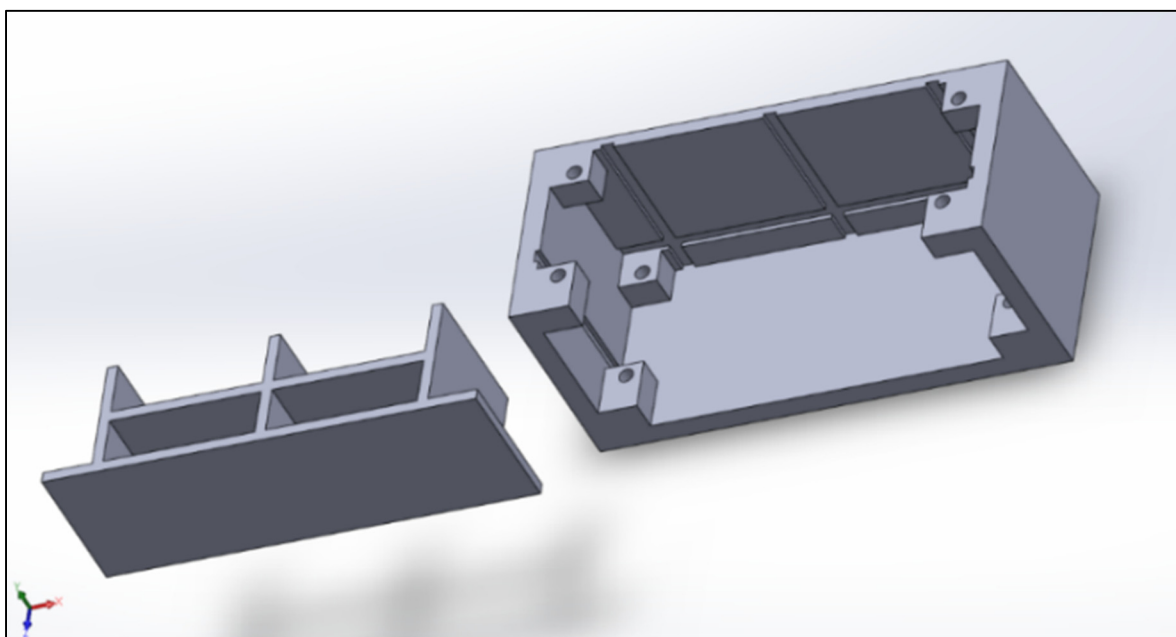


Figure 4.1 Boîtier du banc d'essai

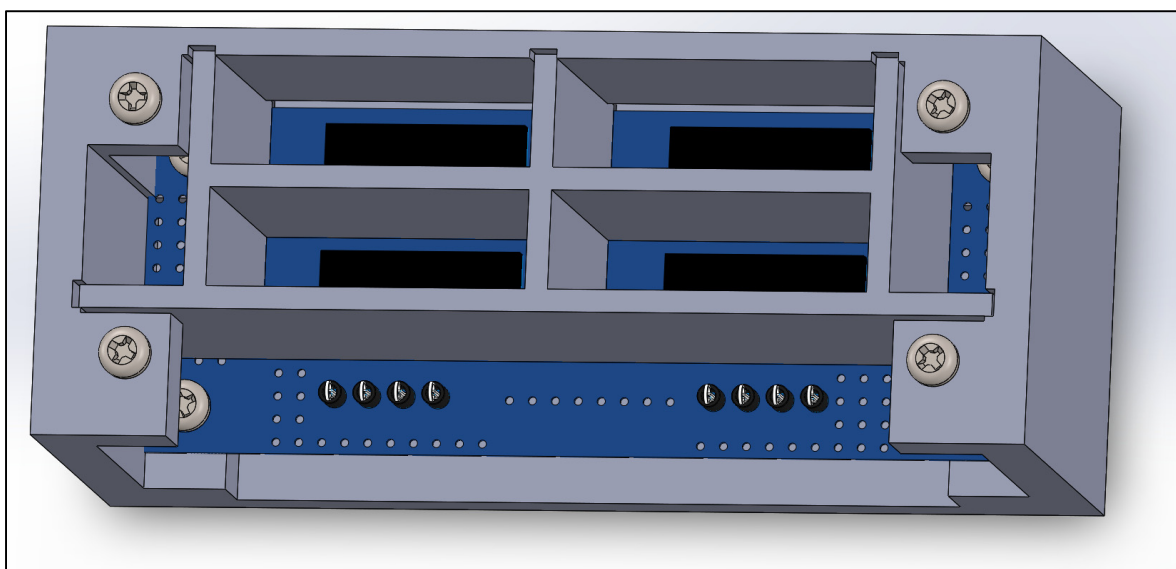


Figure 4.2 Modèle 3D de l'assemblage du boîtier, du PCB de cellules PV, et du mur de séparation

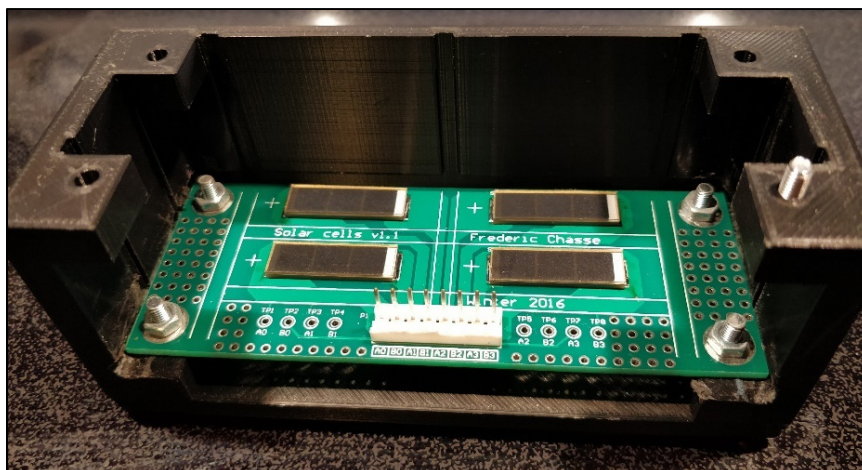


Figure 4.3 PCB de cellules PV encapsulé dans le boîtier

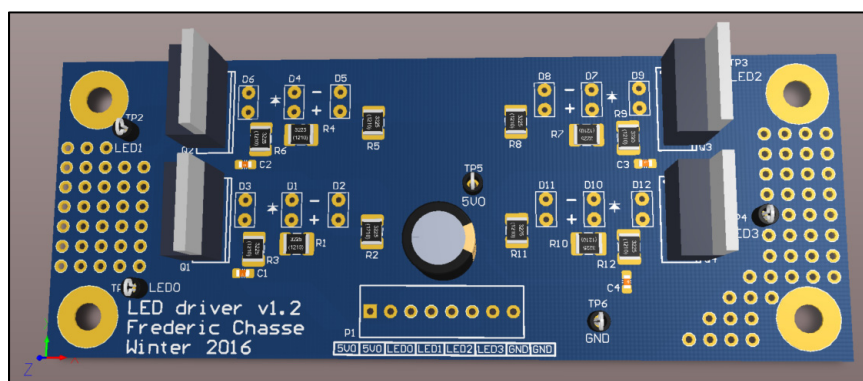


Figure 4.4 Vue 3D du dessus du PCB de DEL



Figure 4.5 Vue du dessous du PCB de DEL du banc d'essai

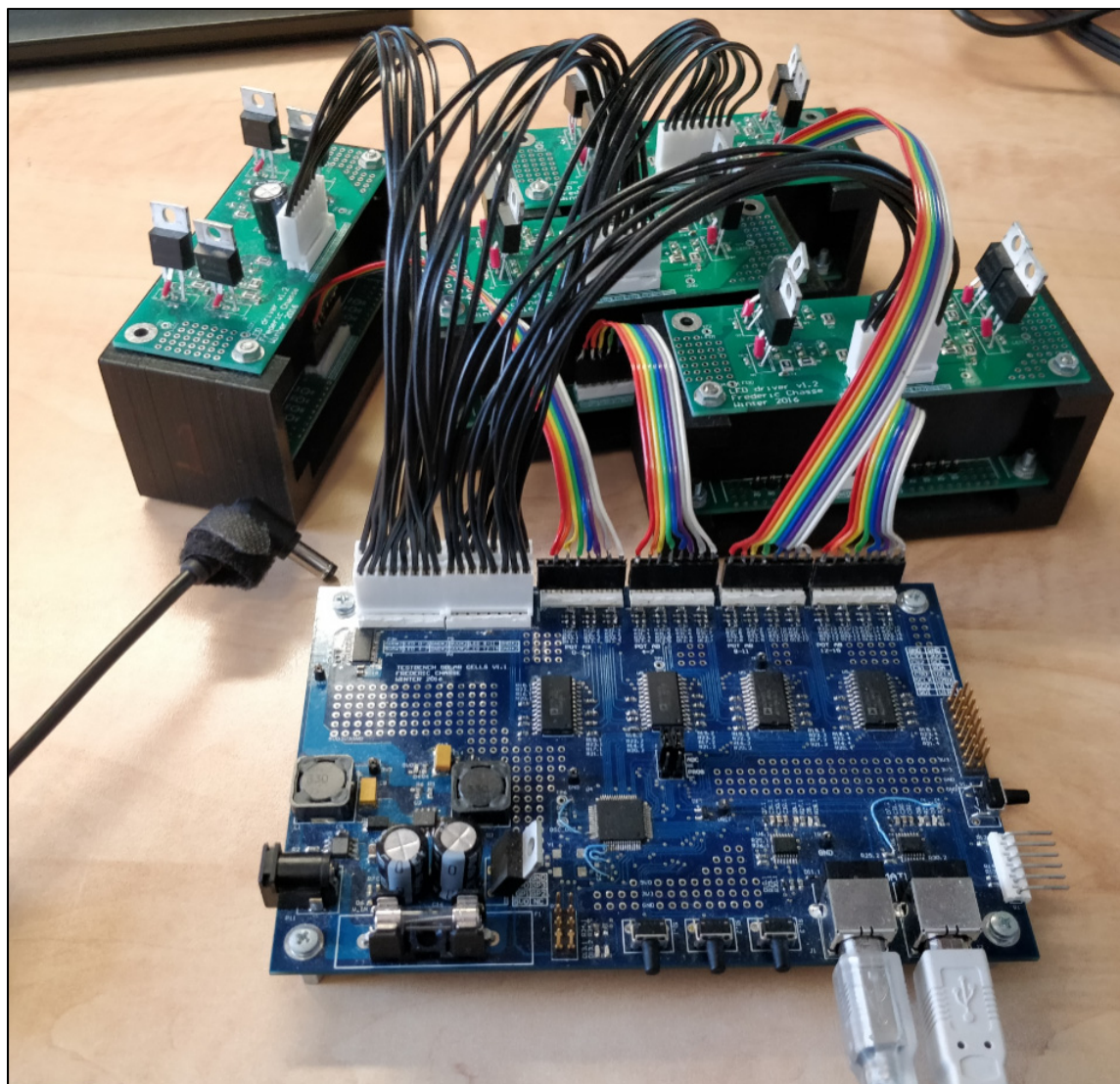


Figure 4.6 Montage complet du banc d'essai

Le PCB de cellules solaires (modèle KXOB22-04X3L d'IXYS) n'est en fait qu'un circuit imprimé où les cellules sont soudées et où un connecteur permet de relier les bornes positive et négative des cellules aux entrées analogiques du microcontrôleur. Le schéma électrique et le dessin du PCB sont donnés à l'ANNEXE VII. Le PCB de DEL est un circuit de MOSFET permettant de permuter rapidement l'alimentation des DEL (modèle VLSC5830 de VISHAY, couleur rouge) avec un signal PWM fourni par le contrôleur de DEL. Le schéma électrique et le dessin du PCB sont fournis à l'ANNEXE VIII. Les dessins 3D du montage mécanique sont donnés à l'ANNEXE IX.

4.2.2 Méthode d'acquisition de données et opération des cellules PV

Les cellules PV ont chacune leur potentiomètre respectif, et les connections sont faites de façon identique aux PCM, tel qu'illustré à la Figure 3.16. Le microcontrôleur commande la résistance externe de chaque cellule et lit leur tension de sortie avec l'ADC. Ensuite, l'équation (3.1) est utilisée pour calculer la puissance de sortie des cellules PV. L'algorithme d'optimisation choisi opère continuellement et envoie ses données à MATLAB par un port série. L'algorithme cesse d'opérer lorsque l'ordinateur envoie la commande appropriée. Il est donc possible d'automatiser une suite de tests et d'en récolter les résultats.

Contrairement aux PCM, les cellules PV ont une dynamique très rapide. Ainsi, le temps d'acquisition a pu être considérablement réduit comparativement à celui utilisé pour les tests sur les PCM. La Figure 4.7 montre la dynamique d'une cellule PV lorsqu'elle reçoit la lumière d'une DEL pulsée par PWM. On voit que la cellule PV prend environ 700 microsecondes pour passer de 0 V à 0.3957 V de manière stable, et environ 70 microsecondes pour passer de 0.3945 V à 0 V. Il est à noter que le premier sommet de 0.3706 V a été atteint en 20 microsecondes, et qu'il a ensuite fallu 680 microsecondes à la cellule PV pour se stabiliser. Comparativement à la PCM, on voit que la dynamique d'une cellule PV est presque instantanée.

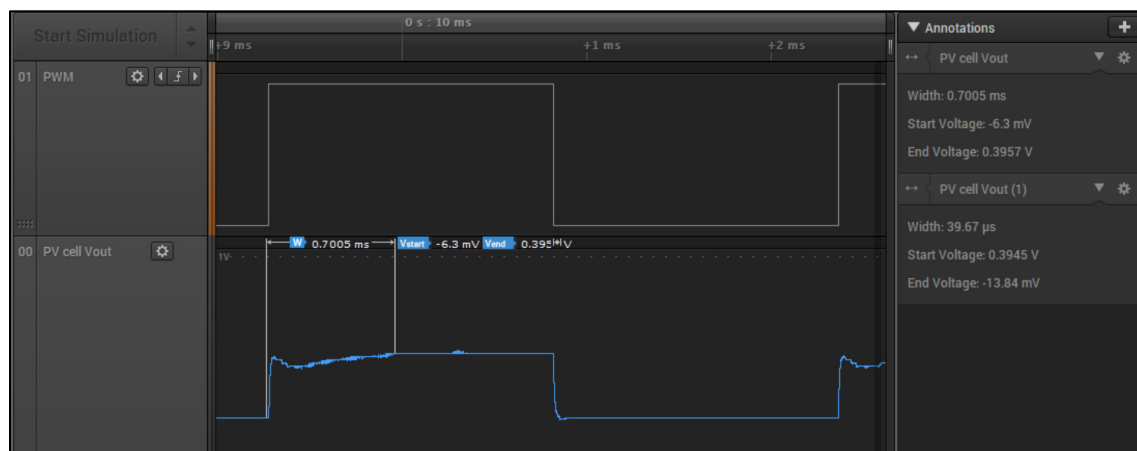


Figure 4.7 Dynamique d'une cellule PV

De plus, étant donné que l'intensité lumineuse reçue par les cellules PV est en fait un signal PWM passant de 0% d'intensité à 100% d'intensité selon un facteur d'utilisation (« duty cycle » en anglais), la tension aux bornes des cellules sera aussi alternée entre 0% et 100%. On peut observer, sur la Figure 4.8, le comportement d'une cellule PV lorsqu'elle est pulsée à un facteur d'utilisation de 30%, et la Figure 4.9 montre le comportement de la même cellule lorsqu'elle est pulsée à 50%. On observe que la tension maximale est approximativement la même pour les deux. Ainsi, pour obtenir la puissance moyenne générée, il faut acquérir plusieurs échantillons et trouver la valeur moyenne de la tension aux bornes de la cellule.



Figure 4.8 Tension de sortie d'une cellule PV avec un cycle de fonctionnement de 30%

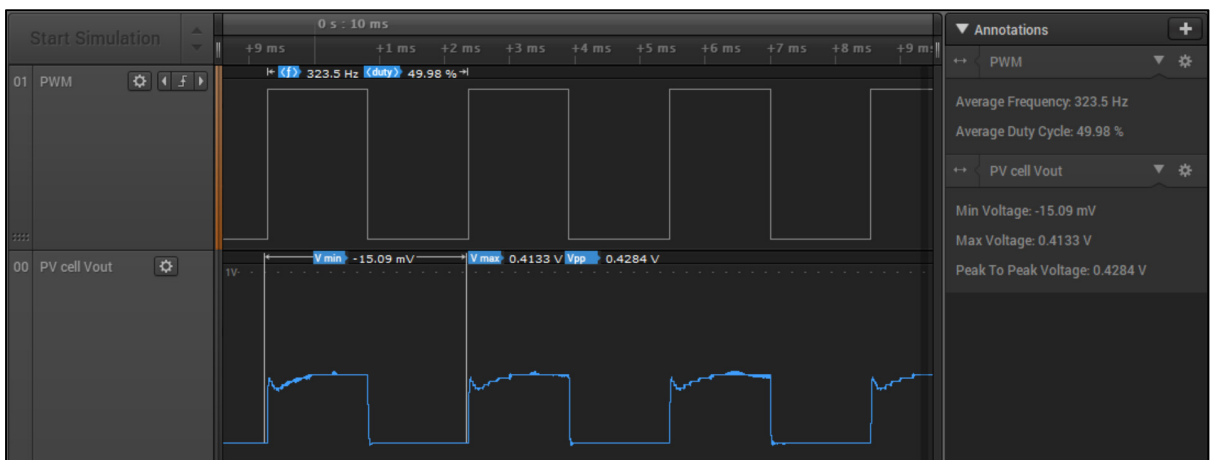


Figure 4.9 Tension de sortie d'une cellule PV avec un cycle de fonctionnement de 50%

La fréquence de fonctionnement choisie du contrôleur de DEL est de 323.5 Hz, ce qui se traduit en une période de 3.0912 ms. Pour obtenir la tension moyenne de sortie des cellules, le microcontrôleur acquiert un échantillon sur ses 15 entrées analogiques à chaque 45 microsecondes, et 600 échantillons par cellule PV sont obtenus. De ces 600 échantillons, les premier et dernier fronts montants sont détectés, et une moyenne est faite sur les échantillons inclus dans la fenêtre délimitée par ces fronts montants. Ainsi, 68.69 échantillons par cycle PWM sont acquis, et de 7 à 8 cycles sont utilisés pour mesurer la tension moyenne des cellules PV. Le temps total d'acquisition pour une itération est de 27 millisecondes, et le temps de traitement après une acquisition est d'environ 5 millisecondes, ce qui amène le temps total d'une itération d'algorithme à 32 millisecondes. À des fins de simplification, le temps de traitement des algorithmes ne sera pas pris en compte, étant donné que ce temps varierait selon le processeur et l'algorithme utilisés. De plus, le code utilisé pour trouver les premier et dernier fronts montants des signaux est ce qui prend le plus de cycles de processeur, et ne serait normalement pas utilisé.

En outre, il a été observé pendant les tests qu'un essai effectué dès le démarrage du système donnait de meilleurs résultats qu'un essai fait quelques minutes après le démarrage. Cela s'explique par le fait que l'intensité lumineuse des DEL fait varier la température des cellules solaires. Or, la température a un certain effet sur la courbe statique de puissance d'une cellule PV (Radziemska, 2003). En effet, une augmentation de la température cause une diminution de la puissance de sortie de la cellule. Bien que cet effet soit négligeable par rapport à l'effet de l'irradiance reçue par la cellule (Kebir, Woodward et Akhrif, 2018), des mesures ont été prises afin d'assurer l'uniformité des résultats. Tout d'abord, les tests sur le banc d'essai sont effectués seulement après que le système ait fonctionné pendant 20 minutes, pour que la température des cellules PV ait le temps de se stabiliser. En outre, les tests ont été effectués selon l'ordre suivant : P&O, OEPS, OEPPC. Dix tests ont été effectués selon cet ordre, et les résultats ont été moyennés. Ces résultats sont présentés à la section 4.3.

Il est aussi nécessaire de noter que, normalement, pour déterminer l'intensité lumineuse reçue par une cellule PV, on parlerait de niveau d'irradiance reçue, en W/m^2 . Cependant, la seule

méthode qui nous permettait de déterminer l'irradiance reçue par les cellules PV était de mesurer la tension en circuit ouvert des cellules PV et de se référer à leur fiche technique. Cependant, nous savons qu'un cycle de fonctionnement des DEL plus haut se traduit en une irradiance reçue par les cellules PV plus haute. Ainsi, à des fins de simplification, nous référons à l'intensité lumineuse reçue par les cellules PV comme étant le cycle de fonctionnement des DEL, et cette valeur est notée IL , pour « intensité lumineuse ».

4.3 Résultats

Cette section donne les résultats des différents cas d'étude effectués sur les cellules PV. Les algorithmes P&O, OEPS, et OEPPC sont utilisés et leurs performances respectives sont comparées. Les cas d'étude sont les suivants : optimisation de 15 cellules PV sans perturbations, et optimisation de 15 cellules PV avec perturbations, où les perturbations sont en fait une variation de l'irradiance reçue par les cellules PV. Les critères de comparaison sont les suivants : le temps nécessaire (en nombre d'itérations) pour que les valeurs minimales et maximales de la variable de contrôle d'une unité ne varient que de $\pm 1\%$, noté T_{conv} , la puissance moyenne (en mW) obtenue à partir de T_{conv} , notée P_{conv} , et l'énergie totale récoltée (en millijoules), notée E_{total} .

4.3.1 Optimisation de cellules PV sans perturbations

Pour ce cas d'étude, 15 cellules PV sont utilisées. L'intensité lumineuse reçue (IL) par toutes les cellules PV est d'un niveau de 50%. Les courbes statiques de puissance des cellules PV à ce niveau d'intensité lumineuse sont données à la Figure 4.10. Le test opère pendant 900 itérations, ou 24.3 secondes, puis s'arrête.

Le Tableau 4.1 donne les paramètres du P&O utilisés pour les tests, le Tableau 4.2 donne les paramètres de l'OEPS utilisés, et le Tableau 4.3 donne les paramètres de l'OEPPC utilisés. La Figure 4.11 montre une optimisation par P&O pour ce cas d'étude, la Figure 4.12 montre une optimisation par OEPS, et la Figure 4.13 montre une optimisation par OEPPC.

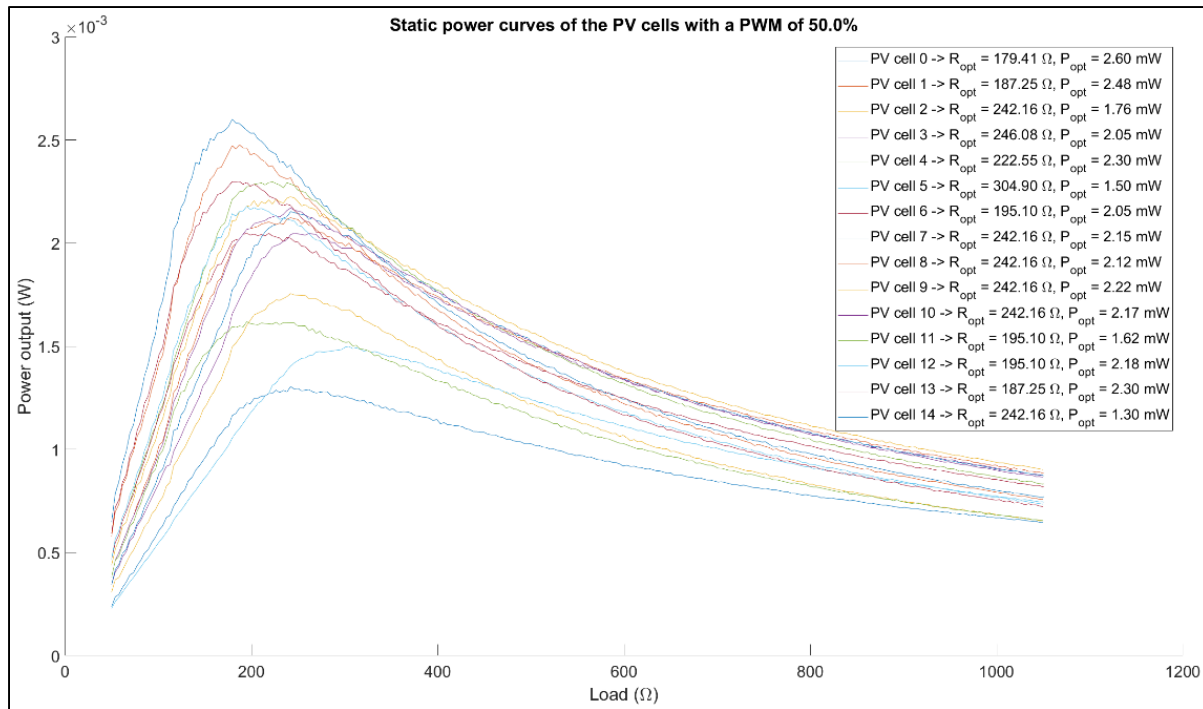


Figure 4.10 Courbes statiques de puissance de 15 cellules PV ayant une intensité lumineuse de 50%

Tableau 4.1 Paramètres de P&O utilisés pour les expérimentations

Paramètre	Valeur	Paramètre	Valeur
Δ	7.8431 Ω	Position minimale	50 Ω
Position initiale	199.0196 Ω	Position maximale	402.9412 Ω

Tableau 4.2 Paramètres de l'OEPS utilisés pour les expérimentations

Paramètre	Valeur	Paramètre	Valeur
c_1	0.5	Marge des sentinelles	$\pm 5\%$
c_2	1.2	Position minimale	50 Ω
ω	0.3	Position maximale	402.9412 Ω
Nombre de particules	5		

Tableau 4.3 Paramètres de l'OEPPC utilisés pour les expérimentations

Paramètre	Valeur	Paramètre	Valeur
c_1	0.5	Marge du classificateur	$\pm 7.8431 \Omega$
c_2	1.2	Position minimale	50Ω
ω	0.3	Position maximale	402.9412Ω
Nombre minimal de particules	3	Δ_1 (P&O)	11.7647Ω
Marge des sentinelles	$\pm 5\%$	Δ_2 (P&O)	3.9216Ω
Nombre d'itérations définissant la convergence pour l'OEPP	5	Amplitude de convergence pour l'OEPP	$\pm 5\%$
Nombre d'itérations définissant la convergence pour le P&O	12	Amplitude de convergence pour le P&O	$\pm 2\Delta$

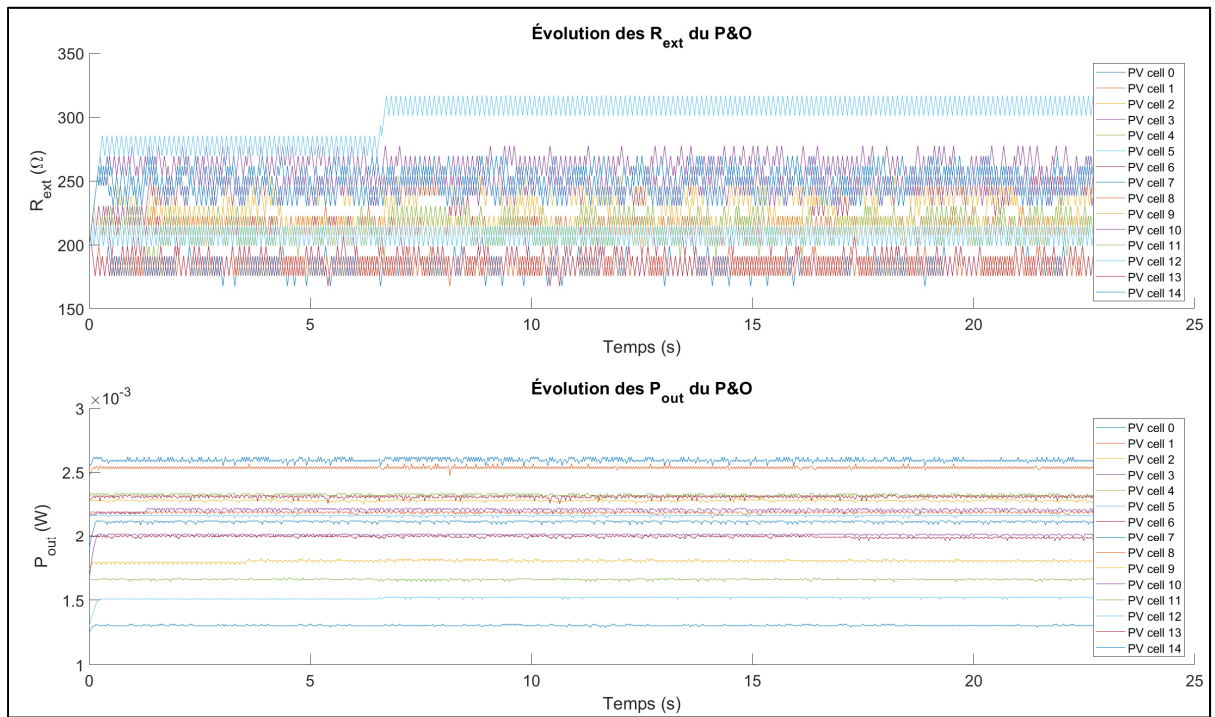


Figure 4.11 Optimisation par P&O de 15 cellules PV sans perturbations

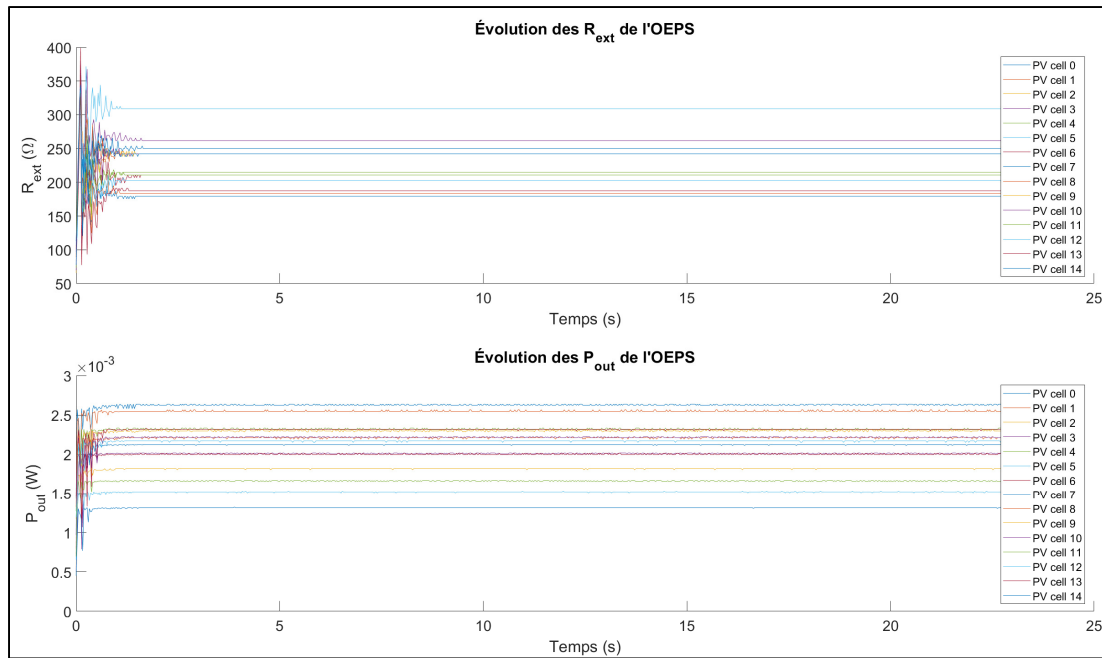


Figure 4.12 Optimisation par OEPS de 15 cellules PV sans perturbations

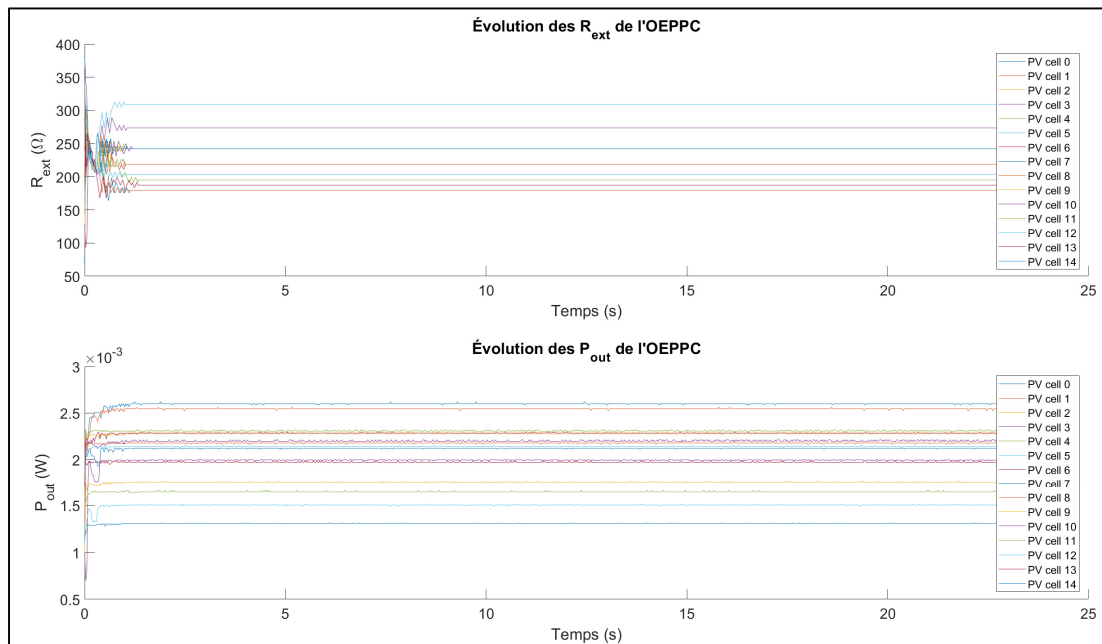


Figure 4.13 Optimisation par OEPPC de 15 cellules PV sans perturbations

On remarque que les puissances de sortie des cellules PV optimisées par P&O semblent beaucoup plus bruitées que lorsque les cellules sont optimisées par OEPS ou par OEPPC. En

effet, bien que le P&O réussisse bien à trouver l'optimum des unités, la nature de l'algorithme le fera osciller autour de ces points optimaux, réduisant ainsi la puissance de sortie moyenne totale. De plus, on remarque que l'approche P&O est plus sensible au bruit de mesure. On voit cet effet sur la cellule PV 5 de la Figure 4.11, où l'algorithme a changé le point d'opération de la cellule vers la 7^e seconde. On peut expliquer ce changement en observant la courbe de puissance de cette cellule PV sur la Figure 4.10. On voit que, bien que la courbe de puissance de cette cellule n'ait qu'un optimum, la courbe de puissance est plus plate autour de ce point. Ainsi, le P&O tentera de rester dans cette région générale, et une mesure bruitée peut amener l'algorithme à un autre point d'opération.

En comparant les optimisations par OEPS et par OEPPC, on peut observer rapidement que les temps de convergence sont assez semblables. Cependant, il est peut-être difficile de le voir sur la courbe de l'OEPPC, mais les unités ayant convergé initialement au bon point avec la partie OEPP de l'algorithme OEPPC sont restées à leur point d'opération, et la convergence de ces unités a été beaucoup plus rapide.

Tableau 4.4 Résultats de l'optimisation de 15 cellules PV sans perturbations

Itérations	Critère	P&O	OEPS	OEPPC
[1, 900] $IL = 50\%$	E_{total} (mJ)	744.9	746.6	747.9
	T_{conv} (itérations)	745.2	48.8	33.2
	P_{conv} (mW)	30.70	30.83	30.85

Le Tableau 4.4 donne les résultats des trois optimisations pour ce cas d'étude. Les meilleurs résultats ont été mis en gras et en italique. On observe que, pour le même système non perturbé, l'OEPPC a récolté plus d'énergie, convergé plus rapidement, et obtenu une puissance de sortie moyenne plus haute que les deux autres algorithmes. Cet écart de performance s'explique par le fait que l'OEPPC profite de la vitesse de convergence très rapide de l'OEPP, et de la capacité d'affinage d'optimum du P&O. Ainsi, on peut affirmer que l'OEPPC présente de meilleures performances que les deux autres algorithmes étudiés.

4.3.2 Optimisation de cellules PV avec perturbations

Pour ce cas d'étude, 15 cellules PV sont utilisées. L'intensité lumineuse (IL) reçue par toutes les cellules PV est initialement d'un niveau de 50%. Les courbes statiques de puissance des cellules PV à ce niveau d'intensité lumineuse sont données à la Figure 4.10. Les cellules restent à ce niveau pendant 300 itérations, ou 8.1 secondes, puis une perturbation est appliquée sur chaque cellule PV. Le niveau d'intensité lumineuse reçue par toutes les cellules est à ce moment de 30%. La Figure 4.14 donne les courbes statiques de puissances des cellules PV à ce niveau d'intensité lumineuse. Les cellules restent à ce niveau pendant 300 itérations, puis une nouvelle perturbation est appliquée sur toutes les cellules, ramenant leur niveau d'intensité lumineuse à 50%. L'algorithme opère encore pendant 300 itérations, puis le test est terminé. Au total, l'algorithme opère pendant 900 itérations, ou 24.3 secondes. Les paramètres des algorithmes sont les mêmes que ceux utilisés dans les tests de la section 4.3.1. La Figure 4.15 montre une optimisation par P&O pour ce cas d'étude, la Figure 4.16 montre une optimisation par OEPS, et la Figure 4.17 montre une optimisation par OEPPC.

En observant les courbes de puissance de la Figure 4.14, on remarque que les mesures sont beaucoup plus bruitées lorsque l'intensité lumineuse reçue est de 30% que lorsqu'elle est à 50%. On peut expliquer cela en comparant la Figure 4.8 et la Figure 4.9. En effet, lorsque le cycle de fonctionnement est de 30%, les cellules n'ont pas le temps de se stabiliser, tandis qu'un cycle de fonctionnement de 50% permet aux cellules de stabiliser leur tension de sortie. Ainsi, les valeurs lues à chaque itération sont plus bruitées pour un cycle de fonctionnement de 30% que pour 50%. Ce bruit de mesure peut affecter la stabilité du P&O, comme on l'observe à la Figure 4.15. On observe aussi un temps de convergence plus long pour certaines unités optimisées avec l'OEPPC à la Figure 4.17. En effet, comme l'OEPPC affine la position finale des unités avec le P&O, il va de soi que des mesures bruitées affecteraient le temps de convergence de l'algorithme. On voit sur la Figure 4.16 que l'OEPS, quant à elle, ne voit pas vraiment de changement sur son temps de convergence.

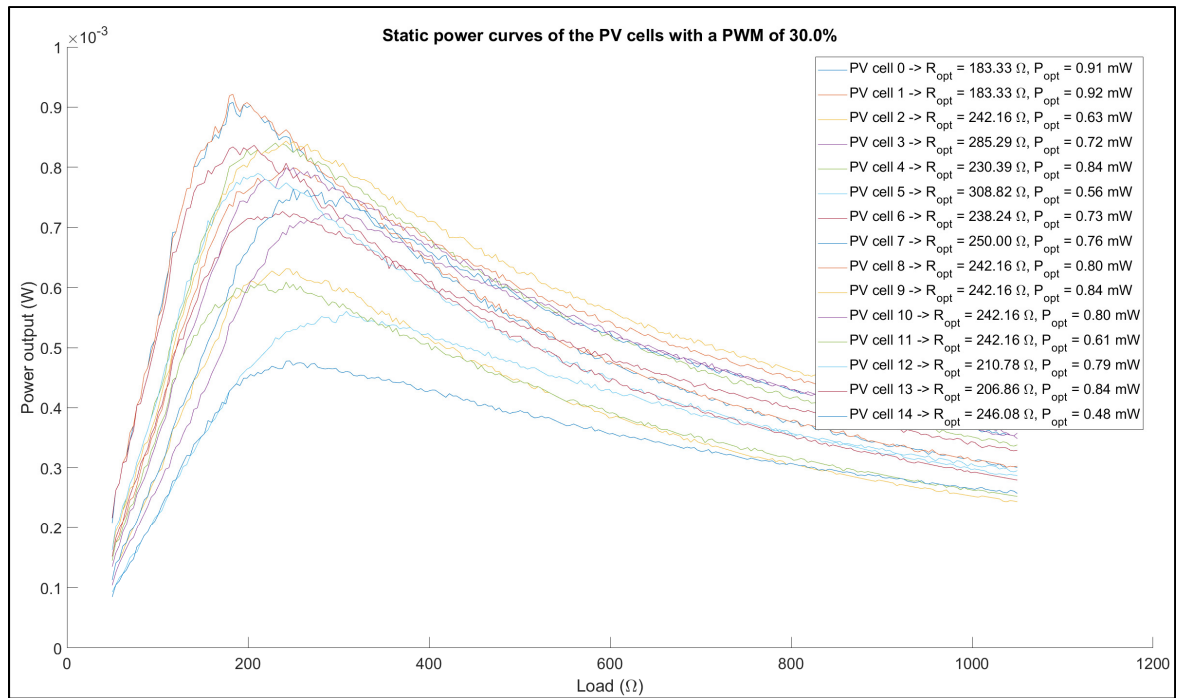


Figure 4.14 Courbes statiques de puissance de 15 cellules PV ayant une intensité lumineuse de 30%

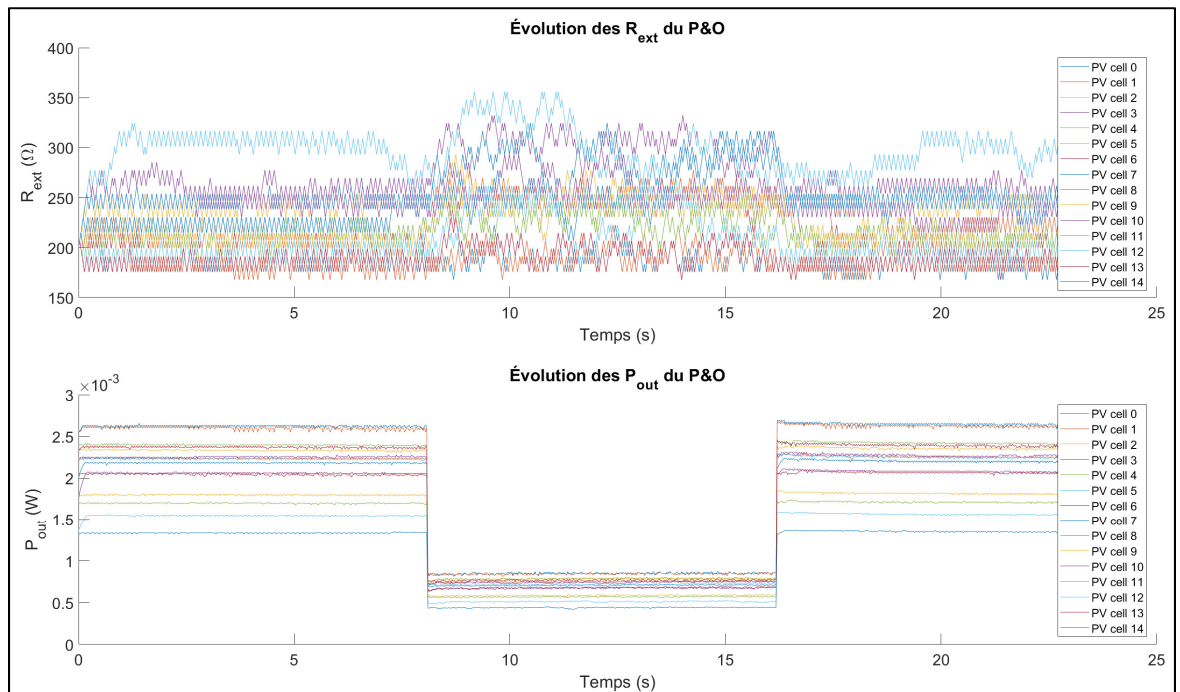


Figure 4.15 Optimisation par P&O de 15 cellules PV avec perturbations sur toutes les unités

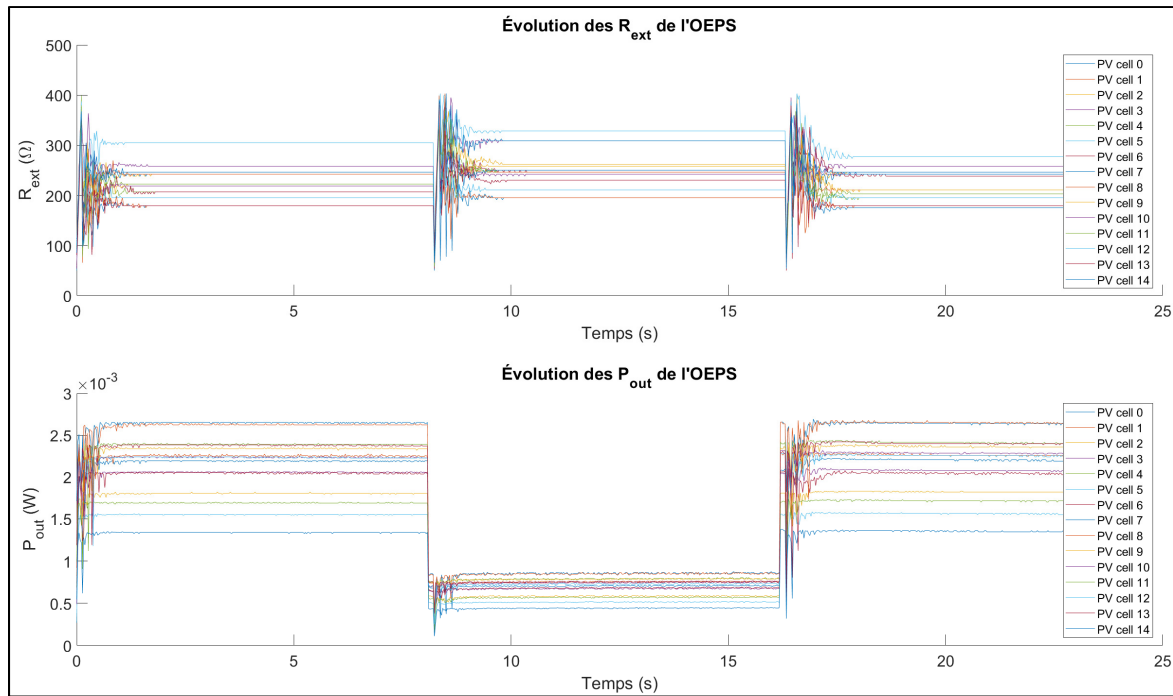


Figure 4.16 Optimisation par OEPS de 15 cellules PV avec perturbations sur toutes les unités

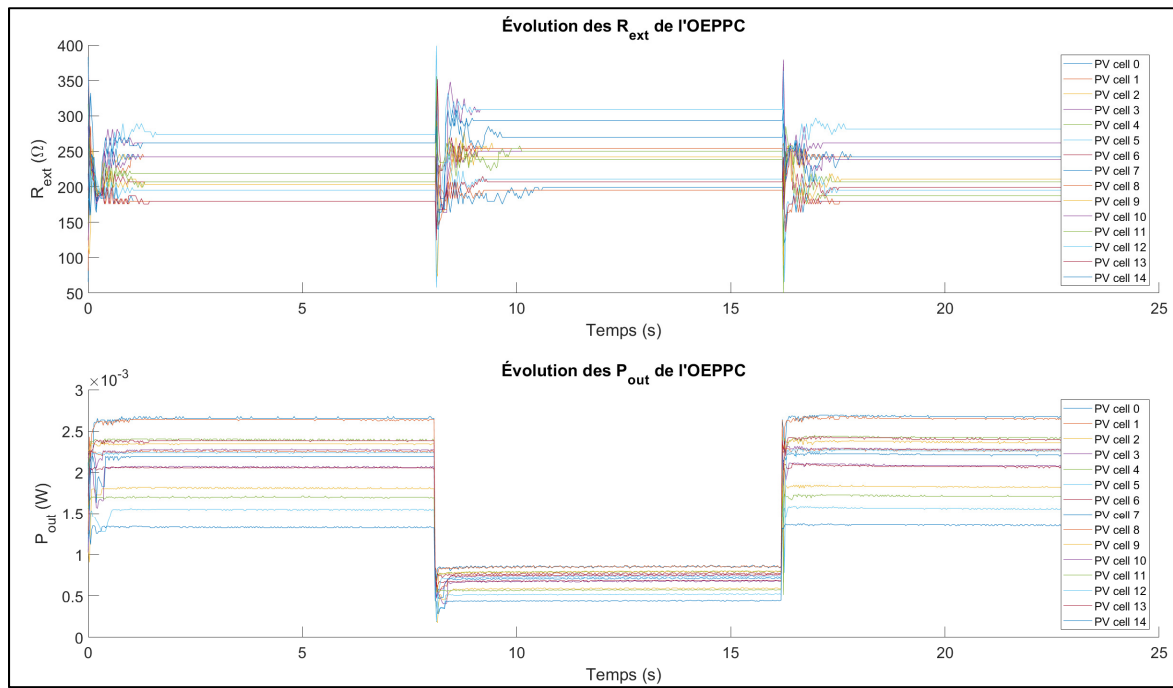


Figure 4.17 Optimisation par OEPPC de 15 cellules PV avec perturbations sur toutes les unités

Le Tableau 4.5 donne les résultats des trois algorithmes pour ce cas d'étude. Comme dans le cas d'étude précédent, on observe que l'OEPPC a convergé plus rapidement, récolté plus d'énergie, et obtenu une puissance de sortie finale plus haute que les deux autres algorithmes. On voit ainsi que l'avantage de vitesse de l'OEPPC, couplé à l'affinement des positions finales avec la partie P&O de l'algorithme.

Tableau 4.5 Résultats de l'optimisation de 15 cellules PV avec perturbations

Itérations	Critère	P&O	OEPS	OEPPC
[1, 300] $IL = 50\%$	E_{total} (mJ)	254.1	253.4	255.4
	T_{conv} (itérations)	246.1	49.9	31.7
	P_{conv} (mW)	31.45	31.60	31.71
[301, 600] $IL = 30\%$	E_{total} (mJ)	83.4	83.2	83.7
	T_{conv} (itérations)	282.9	36.2	29.1
	P_{conv} (mW)	10.39	10.37	10.41
[601, 900] $IL = 50\%$	E_{total} (mJ)	256.7	255.9	257.8
	T_{conv} (itérations)	179.5	38.5	23.0
	P_{conv} (mW)	31.70	31.88	31.98

4.4 Conclusion

Ce chapitre présentait une étude comparative des performances des algorithmes P&O, OEPS, et OEPPC lorsqu'appliqués sur un système composé de 15 cellules photovoltaïques. Deux cas d'étude ont été faits, soient : optimisation sans perturbations, et optimisation avec perturbations. Les deux cas d'étude ont démontré que l'OEPPC performe mieux que le P&O et que l'OEPS au niveau de l'énergie récoltée pendant l'optimisation, au niveau de la vitesse de convergence, et au niveau de la capacité à trouver précisément les optima respectifs des cellules PV. Ainsi, bien que l'OEPPC ait été développée dans le but d'être appliquée sur des PCM, ce chapitre démontre que l'algorithme peut aussi être appliqué à d'autres types de système, et même que l'algorithme performe mieux que le P&O et que l'OEPS, deux algorithmes utilisés grandement dans les systèmes photovoltaïques.

Cependant, il faut aussi mentionner que le cas des perturbations lentes, où l'optimum d'une unité change très lentement, n'a pas été étudié. En effet, ce problème ne nous est apparu que lors des tests sur des PCM réelles. Toutefois, il a été avancé au CHAPITRE 3 que l'ajout de diversité à l'OEPPC permettrait d'adapter l'algorithme à ce type de perturbations environnementales, tout en gardant l'avantage de vitesse et la précision de l'algorithme.

Les résultats obtenus dans ce chapitre nous permettent néanmoins d'affirmer que l'OEPPC est un algorithme rapide et efficace pouvant être utilisé pour optimiser des systèmes dynamiques en temps réel. Il demeure cependant vrai que l'OEPPC nécessite au moins trois unités à optimiser pour être fonctionnel. Ainsi, si le système à optimiser ne comporte qu'une ou deux unités, l'OEPPC ne serait pas apte à optimiser le système.

CONCLUSION

L'objectif de ce mémoire était d'évaluer la possibilité d'utiliser des algorithmes basés sur l'OEP pour optimiser en temps réel la puissance de sortie de piles à combustible microbiennes. En outre, un nouvel algorithme basé sur l'OEP a été proposé, et ses performances ont été comparées à celles d'algorithmes déjà éprouvés dans le domaine des énergies renouvelables.

Étant donné que l'algorithme original de l'OEP n'est pas apte à optimiser des systèmes dynamiques, le CHAPITRE 2 a présenté quelques modifications faites à l'algorithme pour l'adapter à ce contexte. Notamment, l'OEPS a été élaborée, de même que l'OEPP. Un nouvel algorithme proposé dans ce mémoire, c'est-à-dire l'OEPPC, a aussi été expliqué. Cet algorithme prend avantage de la vitesse de l'OEPP, mais ajoute un classificateur à l'algorithme et affine les positions finales des particules par P&O.

Le CHAPITRE 3 présentait des résultats de simulations et d'expérimentations de ces algorithmes sur des PCM. Le P&O a été utilisé comme base de comparaison, car cet algorithme a été utilisé sur plusieurs systèmes, dont les PCM.

La première partie du CHAPITRE 3 se concentrait sur les simulations sur des PCM. Deux cas d'étude ont été faits, soient : optimisation de 10 PCM identiques avec perturbations, et optimisation de 10 PCM non identiques avec perturbations. Les résultats ont prouvé que l'OEPS et le P&O performant de manière semblable, mais que l'OEPS est plus précise que le P&O car l'OEPS ne présente aucune oscillation en régime permanent, tandis que la variable de commande du P&O oscille à une amplitude d'au moins $\pm\Delta$. Ainsi, pour une optimisation de plus longue durée sans perturbations, l'OEPS récolterait plus d'énergie que le P&O.

L'OEPP s'est démontrée très rapide, mais lorsque des différences entre les unités asservies ont été introduites, la précision de l'algorithme a grandement chuté. Ainsi, dans un contexte

réel, où il est très difficile de garantir que les unités seront identiques, l'optimisation par OEPP serait très désavantageuse par rapport à l'OEPS et au P&O.

L'OEPPC, quant à elle, a prouvé être l'algorithme convergeant le plus rapidement (à l'exception de l'OEPP lorsque les unités optimisées ont des courbes de puissance identiques) et ayant la meilleure précision. En effet, l'affinage des positions et la classification des unités lors des perturbations ont démontré que l'on peut profiter de la vitesse de l'OEPP tout en ayant la capacité de trouver l'optimum des unités de façon précise.

La deuxième partie du CHAPITRE 3 présentait les résultats expérimentaux des algorithmes appliqués sur trois PCM réelles. Les algorithmes testés étaient l'OEPS, l'OEPP, et le P&O (par manque de temps, il n'a malheureusement pas été possible de tester l'OEPPC sur les PCM). Comme en simulations, l'OEPP a démontré être l'algorithme le plus rapide. Cependant, en tenant compte du fait que les puissances optimales des PCM ont changé au fil du temps, le P&O a été l'algorithme qui a le mieux performé. En effet, un problème qui n'avait pas été relevé en simulations est apparu pendant les expérimentations, c'est-à-dire l'optimisation d'une PCM lorsque son optimum de puissance varie très lentement. Le P&O, par nature, réussit à suivre un optimum qui varie lentement de manière très efficace, contrairement aux algorithmes basés sur l'OEP. Bien que l'OEPPC n'ait pas été testée sur ce système, on peut assumer qu'un optimum bougeant lentement aurait diminué les performances de l'algorithme. Ainsi, dans un tel contexte, le P&O a été plus apte à optimiser en temps réel la puissance de sortie des PCM. En revanche, il a été suggéré qu'ajouter une diversité à l'OEPPC permettrait à l'algorithme de s'adapter lors d'une telle situation, tout en gardant ses avantages par rapport aux autres algorithmes étudiés. Cette suggestion a d'ailleurs été validée par des simulations sur l'OEPPC avec diversité (OEPPCD). Les résultats obtenus (présentés à l'ANNEXE VI) ont permis de constater que l'OEPPCD offrait une meilleure performance que le P&O.

Comme il n'a pas été possible de tester l'OEPPC sur les PCM réelles, le CHAPITRE 4 présentait les résultats expérimentaux de l'optimisation de la puissance de sortie de cellules

photovoltaïques avec les algorithmes P&O, OEPS, et OEPPC. Les résultats ont démontré que l'OEPPC performe mieux que les deux autres algorithmes, tant au niveau de la vitesse de convergence qu'au niveau de la capacité à trouver précisément l'optimum des unités.

En faisant la synthèse de ces résultats, il a été déterminé que, bien que les algorithmes basés sur l'OEP soient meilleurs que le P&O pour l'optimisation de la puissance de sortie de cellules photovoltaïques, ils sont moins aptes à optimiser des PCM lorsque leur puissance de sortie subit des variations très lentes dues à leur dynamique interne. Cependant, les résultats démontrent que des adaptations à l'OEP pourraient potentiellement augmenter les performances de cet algorithme au-delà de celles du P&O. Par exemple, l'OEPPC a été meilleure que le P&O en simulations et en expérimentations sur les cellules PV, et l'ajout de diversité à l'algorithme a permis de l'adapter aux variations lentes que présentent les PCM. Aussi, il reste vrai que, dans le cas où les variations de puissance sont subites et de grande amplitude, les méthodes d'OTR basées sur l'OEP présentent de meilleures performances que celles de l'optimisation par P&O pour l'OTR de la puissance de sortie de PCM.

Il faut tout de même noter que l'OEPPC nécessite au moins trois unités à optimiser pour être efficace. Ainsi, si le système à optimiser ne comporte qu'une ou deux unités, il vaut mieux se tourner vers l'OEPS dans le cas des cellules PV, et vers le P&O dans le cas des PCM. En outre, les expérimentations sur un système PV ont démontré que l'OEPPC est meilleure que le P&O et que l'OEPS, deux algorithmes largement utilisés pour l'optimisation de la puissance de sortie de systèmes photovoltaïques. L'OEPPC serait donc non seulement intéressante pour les PCM et les systèmes PV, mais pourrait potentiellement être utilisée dans d'autres types de système du domaine des ÉR, tels que les systèmes d'énergie hydrolienne ou éolienne.

La première contribution de ce mémoire est donc la suivante : un nouvel algorithme de suivi du point de puissance maximal en temps réel pouvant être appliqué à différents types de systèmes dynamiques a été proposé et testé, soit l'OEPPC. Les performances de cet

algorithmes se sont démontrées meilleures que celles de l'approche P&O en simulations sur un système composé de PCM, ainsi qu'en expérimentations sur un système PV.

La deuxième contribution de ce mémoire est l'application des méthodes basées sur l'OEP pour l'optimisation en temps réel de la puissance de sortie de piles à combustible microbiennes. Il a été démontré que l'OEP a beaucoup de potentiel dans ce contexte, mais qu'en raison de la dynamique interne lente des PCM, l'algorithme nécessite des modifications afin d'être adapté à une telle contrainte. Par exemple, l'OEPPCD a été proposée et testée en simulation, et s'avère être plus efficace que l'approche P&O. Cependant, si les variations de puissance que subissent les PCM à être optimisées sont de forte amplitude et sont rapides, comme on l'observe souvent sur les cellules PV, l'OEPS et l'OEPPC sont deux techniques viables pour l'OTR de la puissance de sortie des PCM.

RECOMMANDATIONS

Ce mémoire a été fait en partie dans le but d'évaluer la possibilité d'utiliser l'OEP comme méthode d'MPPT pour les PCM. Ainsi, comme tout travail de recherche, les conclusions tirées de ce travail peuvent être utilisées comme base pour des travaux futurs.

Il serait notamment intéressant de valider expérimentalement l'OEPPC et l'OEPPCD sur des PCM. En effet, il a été prouvé en expérimentations que l'OEPPC est un algorithme efficace pour l'optimisation de cellules PV, mais il a aussi été déterminé en simulations que l'OEPPCD serait plus apte à optimiser des PCM que l'OEPPC. Cependant, dans le cas où les PCM ont une dynamique interne très stable, l'OEPPC performerait mieux que l'OEPPCD. Ainsi, l'étude de l'OEPPC et de l'OEPPCD sur un système réel constituerait un jalon important pour l'optimisation en temps réel de la puissance de sortie de PCM.

De plus, il serait important de remplacer les potentiomètres utilisés dans ce mémoire par de vrais convertisseurs CC/CC. En effet, dans un contexte réel, la puissance de sortie des PCM serait récoltée dans une charge par l'entremise d'un convertisseur CC/CC, ce qui pourrait causer des pertes au niveau de la conversion d'énergie. Ainsi, l'utilisation d'un convertisseur serait un pas de plus vers l'implantation des PCM dans le secteur industriel.

Il a aussi été noté que le temps de convergence d'une PCM varie selon son point d'opération. Donc, en vue d'accélérer la vitesse de convergence d'un algorithme, il serait faisable d'évaluer la stabilité d'une PCM avant de changer son point d'opération. De cette façon, il serait possible de sauver plusieurs minutes, et cette méthode permettrait aussi de s'assurer d'avoir la vraie courbe de puissance statique de la PCM pour l'optimisation de celle-ci.

Finalement, bien que l'OEPPC ait été suggérée dans ce mémoire, il existe une multitude d'autres algorithmes d'optimisation en temps réel grandement utilisés comme techniques d'MPPT, mais qui n'ont pas été appliqués sur les PCM. Ainsi, la comparaison de l'OEPPCD à ces algorithmes, ou même l'étude seule de ces différents algorithmes permettrait

d'augmenter le bassin de connaissances sur l'optimisation en temps réel de la puissance de sortie des PCM.

ANNEXE I

IMPLÉMENTATION MATLAB DU MODÈLE D'UNE PILE À COMBUSTIBLE MICROBIENNE DE PINTO ET AL. (2010)

Modèle d'origine

```
function [dx, Pout] = mfcModel(T, x, dummy, S0, Rext)

%% Initialisation

F = 96485;
R = 8.314472;
T = 298.15;
Y = 22.753;
Ych4 = .3;
qmax_a = 8.48;
qmax_m = 8.2;
umax_a = 1.9753;
umax_m = .1;
Ks_a = 20;
Ks_m = 80;
m = 2;
gamma = 663400;
Mtotal = .05;
Km = .2*Mtotal;
Kd_a = .02*umax_a;
Kd_m = .02*umax_m;
Xmax_a = 512.5;
Xmax_m = 537.5;
Kx = .04;
Rmin = 25;
Rmax = 2000 + Rmin;
Emin = .01;
Emax = 0.25 + 0.4144* Emin;
Kr = .024;

i0ref = 0.4*4;

e = 0.0001*Mtotal;

V = 0.05;
Fin = 0.1527;
```

```

D = Fin/V;

S    = x(1);
xa   = x(2);
xm   = x(3);
Mox  = x(4);

%% Équations dynamiques

Mred = Mtotal - Mox;

Rint = Rmin + (Rmax - Rmin) * exp(-Kr*xa);           % (22)
Eocv = Emin + (Emax - Emin) * exp(-1/(Kr*xa));       % (23)

alpha_a = 1/2*(1 + tanh(Kx*(xa + xm - Xmax_a)));     % (9)
alpha_m = 1/2*(1 + tanh(Kx*(xa + xm - Xmax_m)));     % (9)

qa = qmax_a * S / (Ks_a + S) * Mox / (Km + Mox);     % (15)
qm = qmax_m * S / (Ks_m + S);                       % (16)

ua = umax_a * S / (Ks_a + S) * Mox / (Km + Mox);     % (13)
um = umax_m * S / (Ks_m + S);                       % (14)

n_conc = R*T/F/m * log(Mtotal/Mred);                 % (18)

Imfc = (Eocv - n_conc) / (Rext + Rint) * Mred / (e+Mred); % (21)

Sp = -qa*xa - qm*xm + D*(S0 - S);                   % (6)
Xap = ua*xa - Kd_a*xa - alpha_a*D*xa;               % (7)
Xmp = um*xm - Kd_m*xm - alpha_m*D*xm;               % (8)

Moxp = -Y*qa + gamma*Imfc / (m*F*V*xa) * 86400;     % (12)

if xa < 1 && Xap < 0
    Xap = 0;
end

if xm < 0.99 && Xmp < 0
    Xmp = 0;
end

% Sortie
Pout = Rext*Imfc^2;

dx = [Sp; Xap; Xmp; Moxp];

```

Modèle accéléré

```

function [dy, Pout]= mfcModelFast(T, y, dummy, S0, Rext)
F = 96485;
R = 8.314472;
T = 298.15;
yield = 22.753;
% Ych4 = .3;
qmax_a = 8.484;
qmax_m = 8.2;
umax_a = 1.9753;
umax_m = .1;
Ks_a = 20;
Ks_m = 80;
m = 2;
gamma = 663400;
Mtotal = .05;
Km = .2*Mtotal;
% Kd_a = .02*umax_a;
% Kd_m = .02*umax_m;
Xmax = 512.5;
Kx = .01*4;
Rmin = 25;
Rmax = 2000 + Rmin;
Emin = .01;
Emax = 0.25 + 0.4144+ Emin;
Kr = .024;

% i0ref=0.4*4;

e = 0.0001*Mtotal;

V = 0.2;
Fin = 0.1527;

D = Fin/V;

Xtot = 500;

Xeg = 250;

S = y(1);
Mox = y(2);
xm = Xtot - Xeg;

```

```
%% Équations dynamiques
```

```
Mred = Mtotal - Mox;
```

```
Rint = Rmin + (Rmax - Rmin) * exp(-Kr*Xeg); % (22)
```

```
Eocv = Emin + (Emax - Emin) * exp(-1/(Kr*Xeg)); % (23)
```

```
qa = qmax_a * S / (Ks_a + S) * Mox / (Km + Mox); % (15)
```

```
qm = qmax_m * S / (Ks_m + S); % (16)
```

```
temp = Eocv - (R*T/m/F)*abs(log(complex(Mtotal/Mred)));
```

```
Imfc = temp/(Rext+Rint)*Mred/(e+Mred);
```

```
Pout = (temp*Mred/(e+Mred)-Rint*Imfc)*Imfc;
```

```
Sp = -qa*Xeg - qm*xm + D*(S0 - S); % (6)
```

```
Moxp = -yield*qa + gamma*Imfc / (m*F*V*Xeg) * 86400; % (12)
```

```
dy = [Sp; Moxp];
```

ANNEXE II

ORDINOGRAMME DÉTAILLÉ DU CLASSIFICATEUR DE L'OEPPC

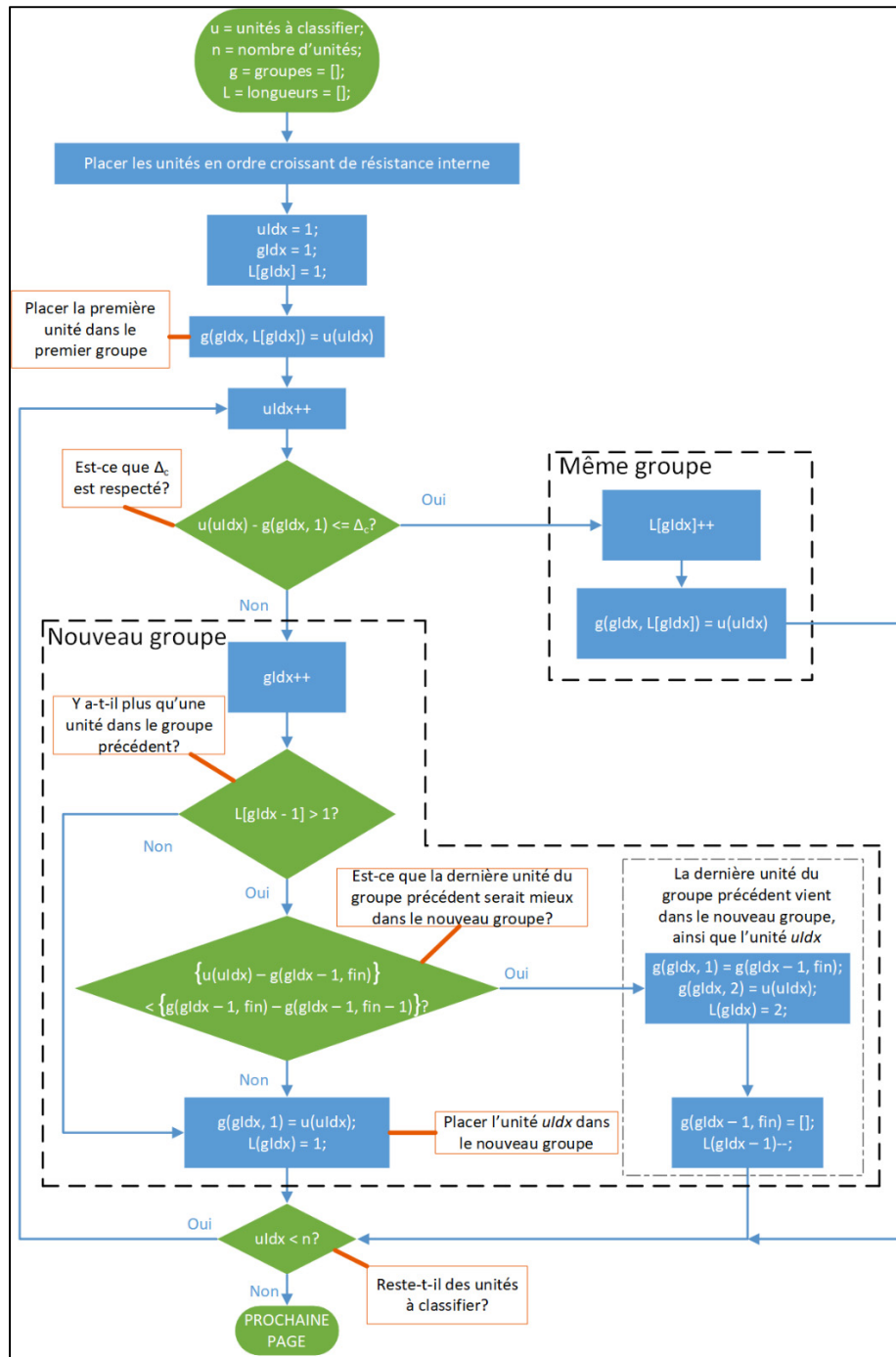


Figure-A II-1 Partie 1 de l'ordinogramme détaillé du classificateur de l'OEPPC

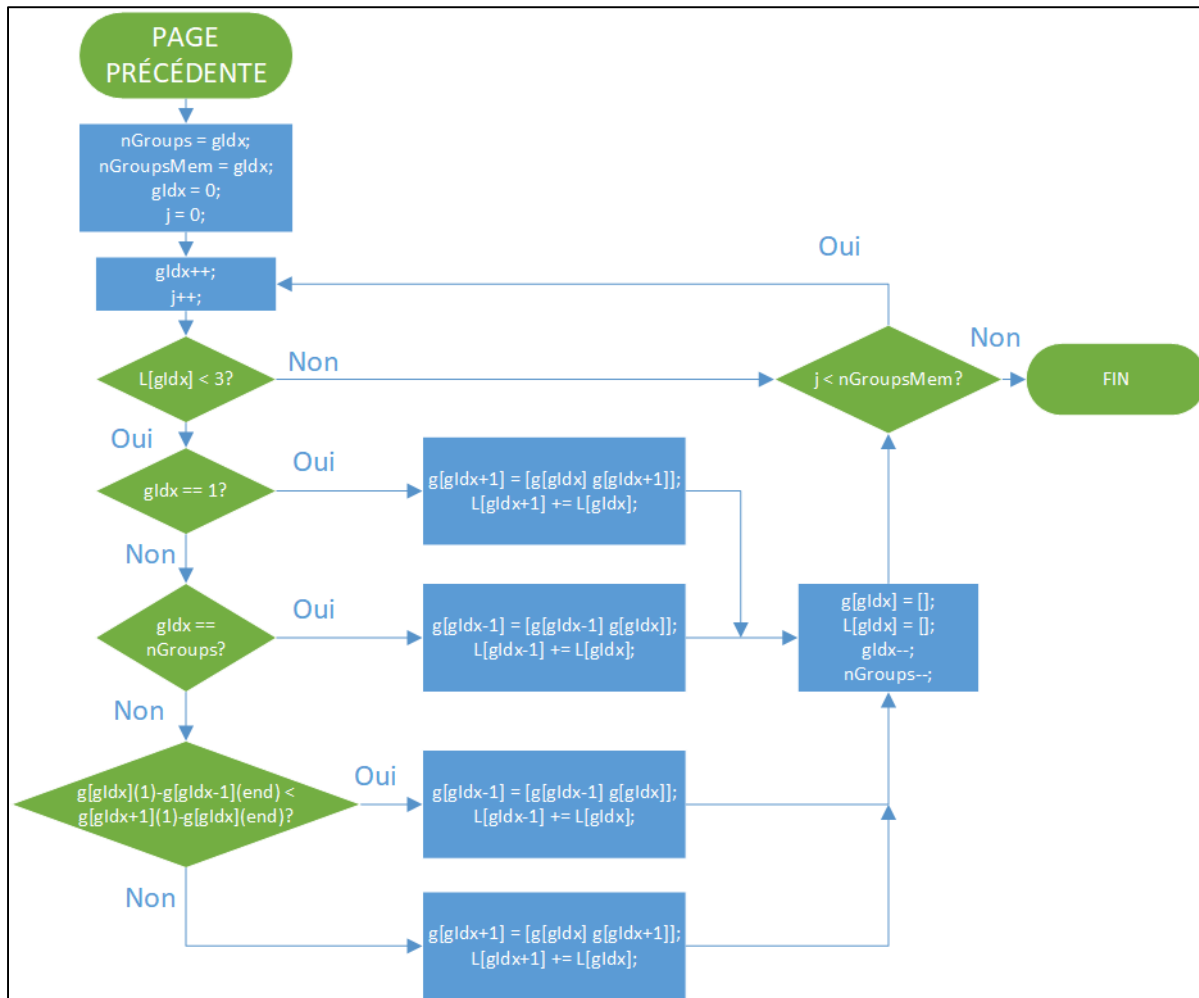


Figure-A II-2 Partie 2 de l'ordinogramme détaillé du classificateur de l'OEPPC

ANNEXE III

SIMULATION DE L'OPTIMISATION DE 10 PCM NON IDENTIQUES AVEC PERTURBATIONS SUR QUELQUES UNITÉS

Dans ce cas d'étude, 10 PCM ayant des courbes statiques de puissance différentes sont utilisées. Les valeurs initiales de concentration de substrat d'alimentation sont : [700, 700, 650, 650, 600, 600, 675, 675, 625, 625] pendant 100 itérations. À la centième itération, une perturbation est appliquée sur le S_0 des cinq premières PCM, où $S_0 = S_0 - 400 \text{ mg/L}$. Ainsi, les S_0 des unités deviennent : [300, 300, 250, 250, 200, 600, 675, 675, 625, 625], et ce pendant 100 itérations. Puis, une nouvelle perturbation de $S_0 = S_0 + 400$ ramène le S_0 de chaque unité déjà perturbée à son S_0 initial pendant 100 itérations. Au total, les algorithmes optimisent les PCM pendant 300 itérations, ou 720 heures. Les courbes de puissance des PCM sont montrées à la Figure 3.7.

Ce cas d'étude est fait pour noter un autre avantage de l'OEPPC par rapport à l'OEPP. En effet, si seulement quelques unités d'un système sont perturbées, l'OEPP redémarrera la recherche pour toutes les unités, tandis que l'OEPPC ne recommencera une recherche que pour les unités perturbées. Il serait en effet fortement désavantageux de redémarrer une optimisation sur des unités qui sont déjà à leur optimum.

La Figure-A III-1 montre l'application du P&O pour ce cas d'étude. On note que seulement les unités perturbées cherchent à se repositionner, ce qui est tout à fait normal, étant donné que le P&O est appliqué individuellement sur chaque unité.

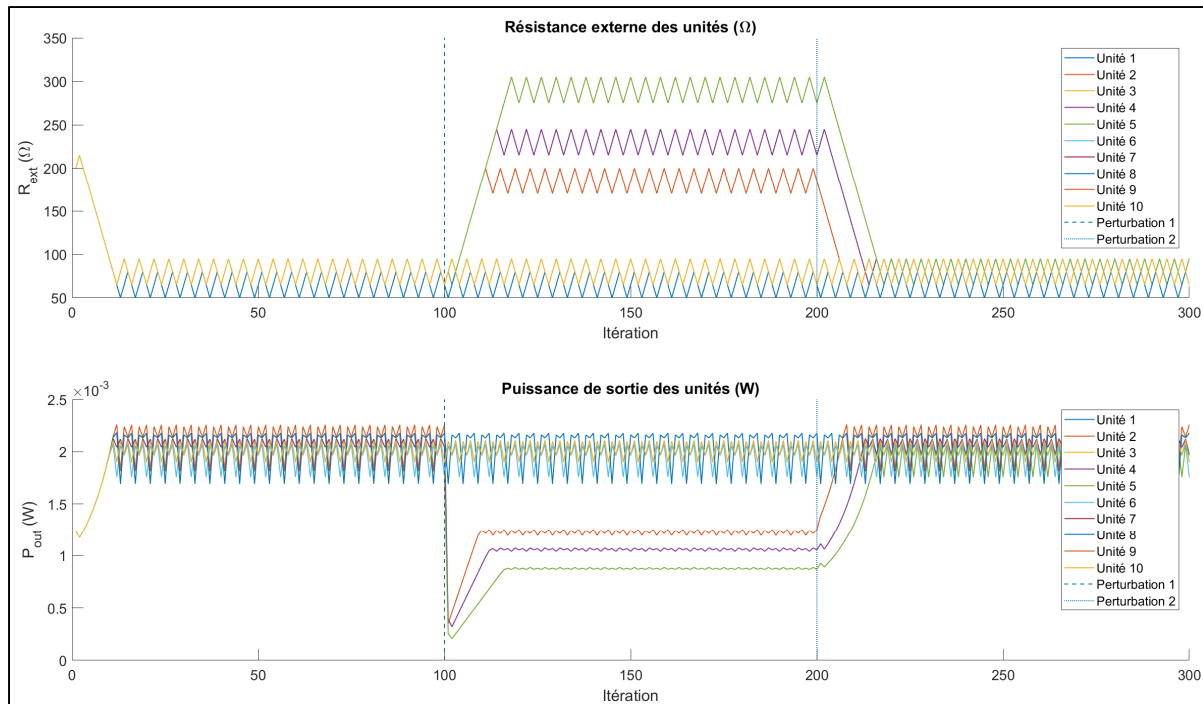


Figure-A III-1 Optimisation par P&O de PCM non identiques
avec perturbations sur quelques unités

La Figure-A III-2 montre une application de l'OEPS pour ce cas d'étude. Tout comme le P&O, on peut voir que seulement les unités perturbées redémarrent une recherche, tandis que les unités non affectées restent à leur optimum.

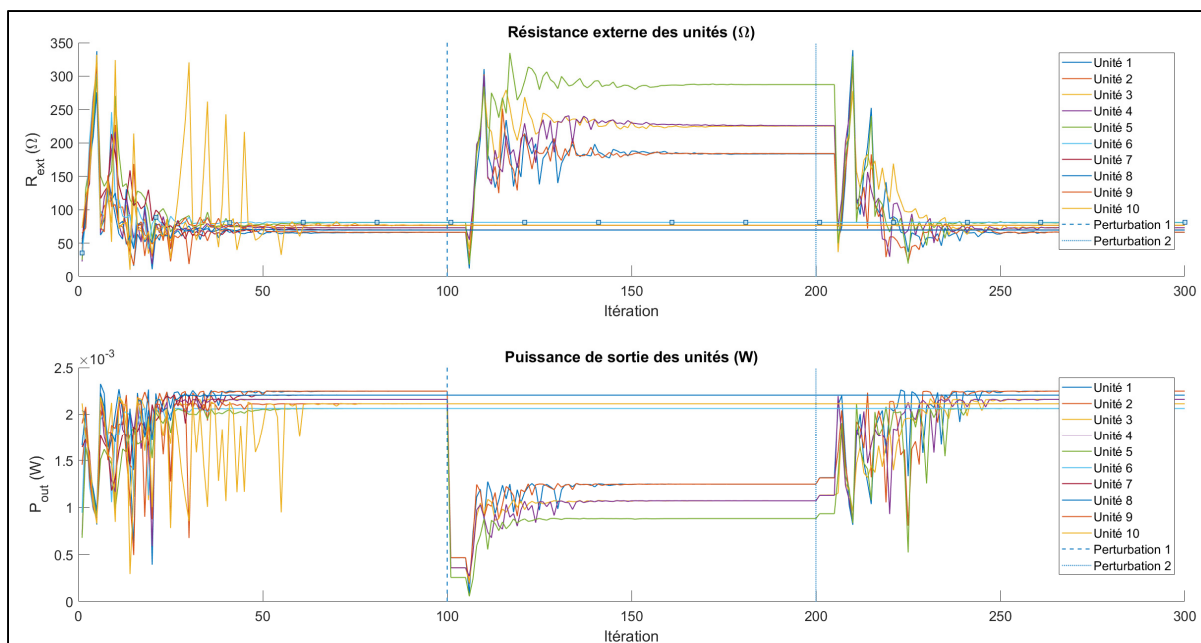


Figure-A III-2 Optimisation par OEPS de PCM non identiques avec perturbations sur quelques unités

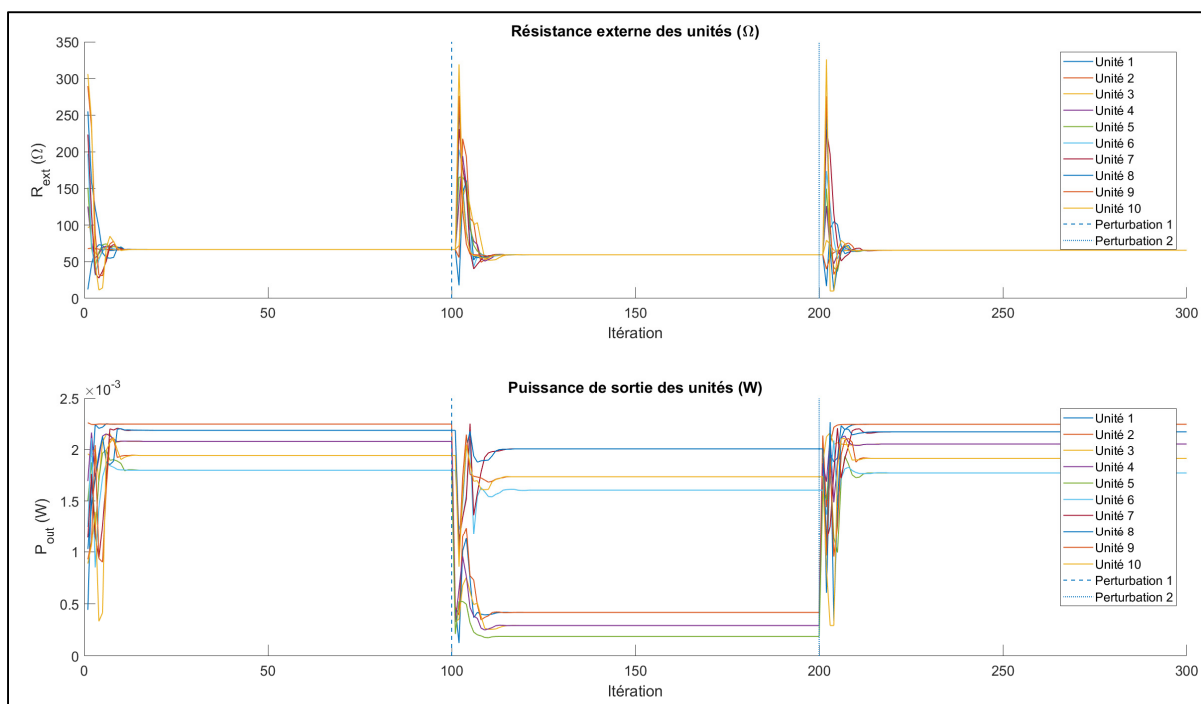


Figure-A III-3 Optimisation par OEPP de PCM non identiques avec perturbations sur quelques unités

La Figure-A III-3 montre l'application de l'OEPP pour ce cas d'étude. On peut observer qu'après une perturbation, une optimisation est redémarrée pour chaque unité. De plus, comme les unités sont différentes, plusieurs unités se retrouvent à une position qui n'est pas leur optimum.

La Figure-A III-4 montre l'application de l'OEPPC pour ce cas d'étude. On observe bien que seulement les unités perturbées sont repositionnées dans l'espace de recherche. De plus, On voit que l'OEPPC a convergé plus rapidement que l'OEPS et de manière à peu près égale au P&O.

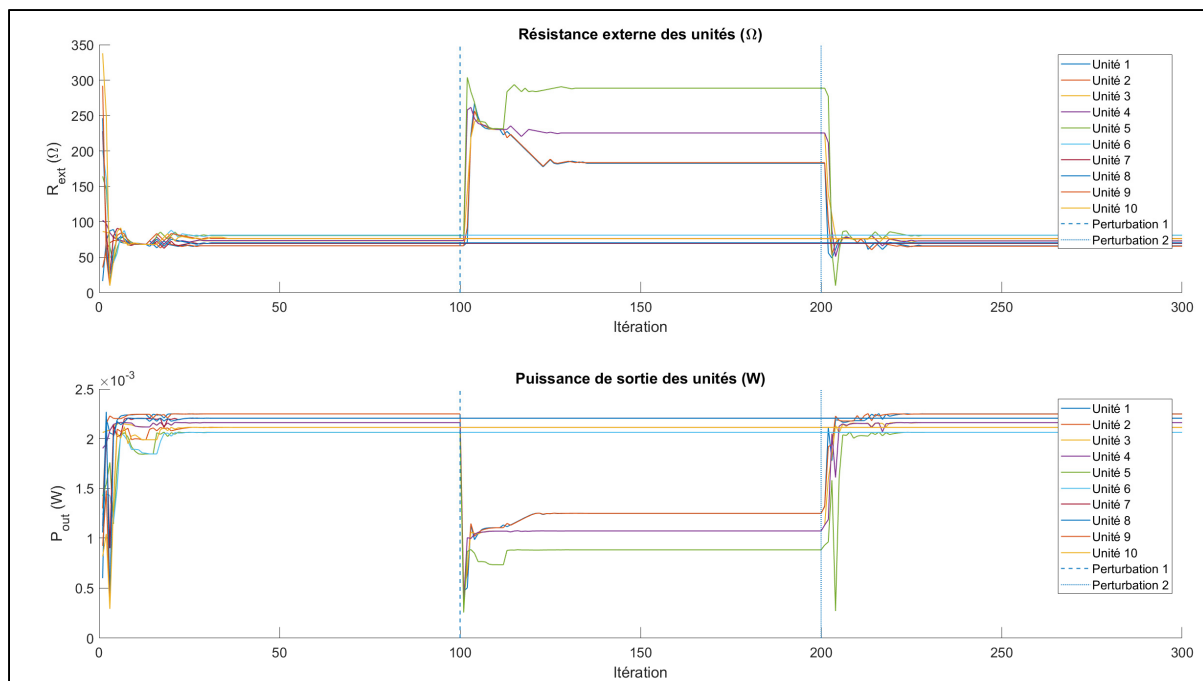


Figure-A III-4 Optimisation par OEPPC de PCM non identiques avec perturbations sur quelques unités

Le Tableau-A III-1 compile les résultats de simulation pour ce cas d'étude. Les résultats sont séparés en trois sections, c'est-à-dire une section par niveau d' S_0 . Les critères de comparaison sont les mêmes qu'à la section 3.2.2.1. Les meilleures performances ont été mises en gras et en italique. Il est à noter que les mesures de vitesse de convergence d'un algorithme ont été faites en faisant la moyenne de chaque unité. Ainsi, ces mesures de

convergence du P&O sont biaisées car les unités n'ayant pas été perturbées étaient déjà à leur optimum. Par exemple, pour les itérations 101 à 200 du P&O, les temps de convergence des unités (en nombre d'itérations) étaient : [10, 10, 13, 13, 17, 1, 2, 2, 1, 1], ce qui donne une moyenne de 7.

On remarque d'abord que l'OEPPC présente le plus grand nombre de joules récoltées pour chaque section. En effet, en ayant une meilleure précision que le P&O et que l'OEPP, et une vitesse de convergence plus rapide que l'OEPS, l'algorithme réussit à récolter plus d'énergie. On peut aussi observer que l'OEPPC a obtenu une puissance moyenne finale égale ou supérieure à l'OEPS, et supérieure aux deux autres algorithmes. On peut expliquer cette supériorité avec le fait que l'OEPPC affine mieux la position finale des unités que l'OEPS grâce à ses essais P&O.

Tableau-A III-1 Résultats de simulation de PCM non identiques avec perturbations sur quelques unités

Itérations	Critère	P&O	OEPS	OEPP	OEPPC
[1, 100]	Joules récoltées	17022.129	17434.545	17510.893	18209.965
	T_{conv}	11.4	62.91	8.36	24.56
	P_{conv}	2.0413	2.1579	2.0601	2.1579
[101, 200]	Joules récoltées	13099.569	13541.983	10490.954	13765.816
	T_{conv}	7	44.35	10.18	14.05
	P_{conv}	1.5535	1.6208	1.2321	1.6217
[201, 300]	Joules récoltées	17204.603	17530.802	17551.951	18290.410
	T_{conv}	6.6	49.69	8.95	13.24
	P_{conv}	2.0419	2.1367	2.0687	2.1579

ANNEXE IV

SCHÉMA ÉLECTRIQUE DE LA CARTE D'ACQUISITION ET DE CONTRÔLE

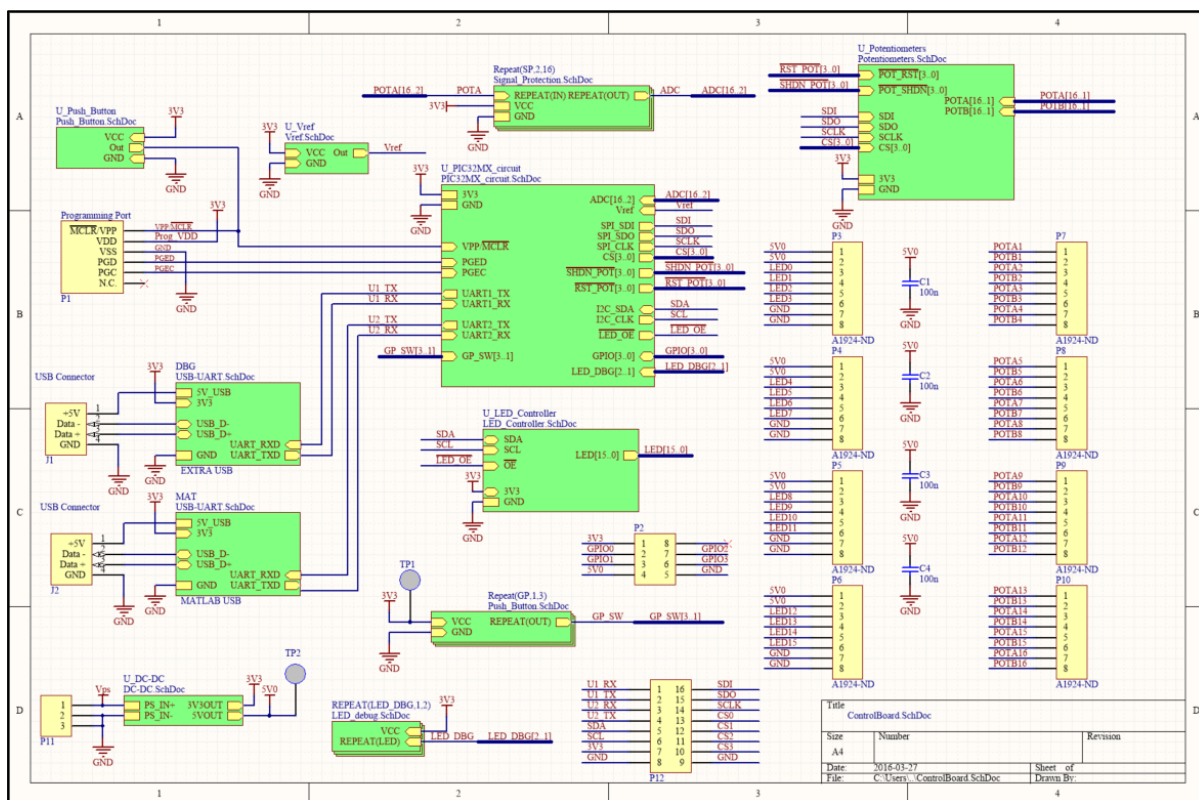


Figure-A IV-1 Schéma bloc du système

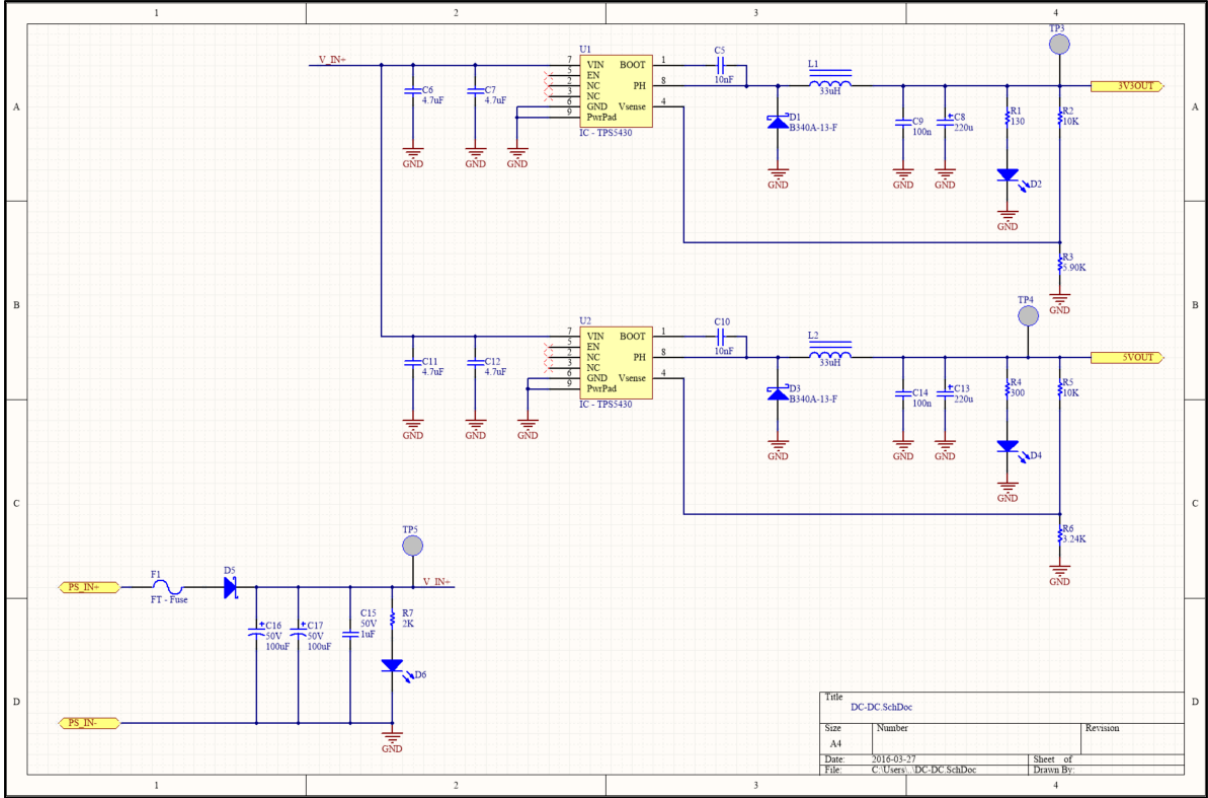


Figure-A IV-2 Circuit d'alimentation du PCB

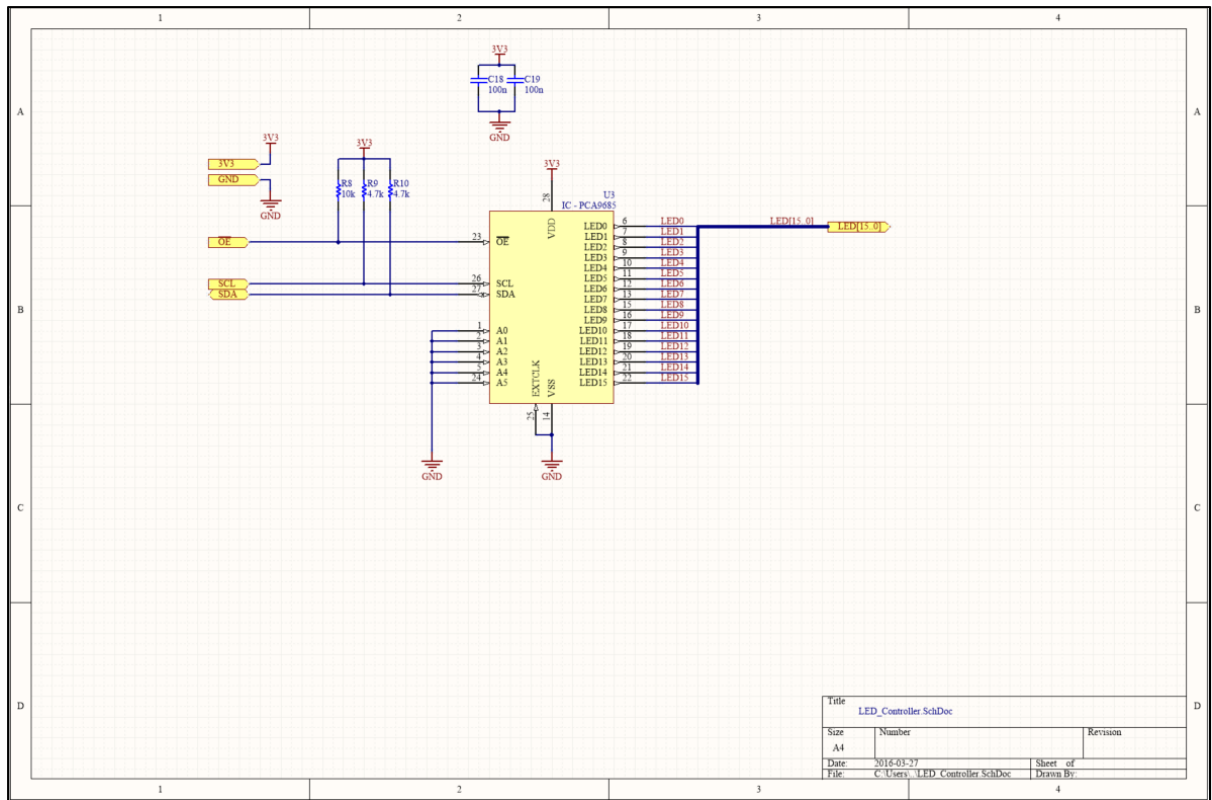


Figure-A IV-3 Circuit de contrôle des DEL pour les cellules PV

Figure-A IV-4 Circuit du microcontrôleur

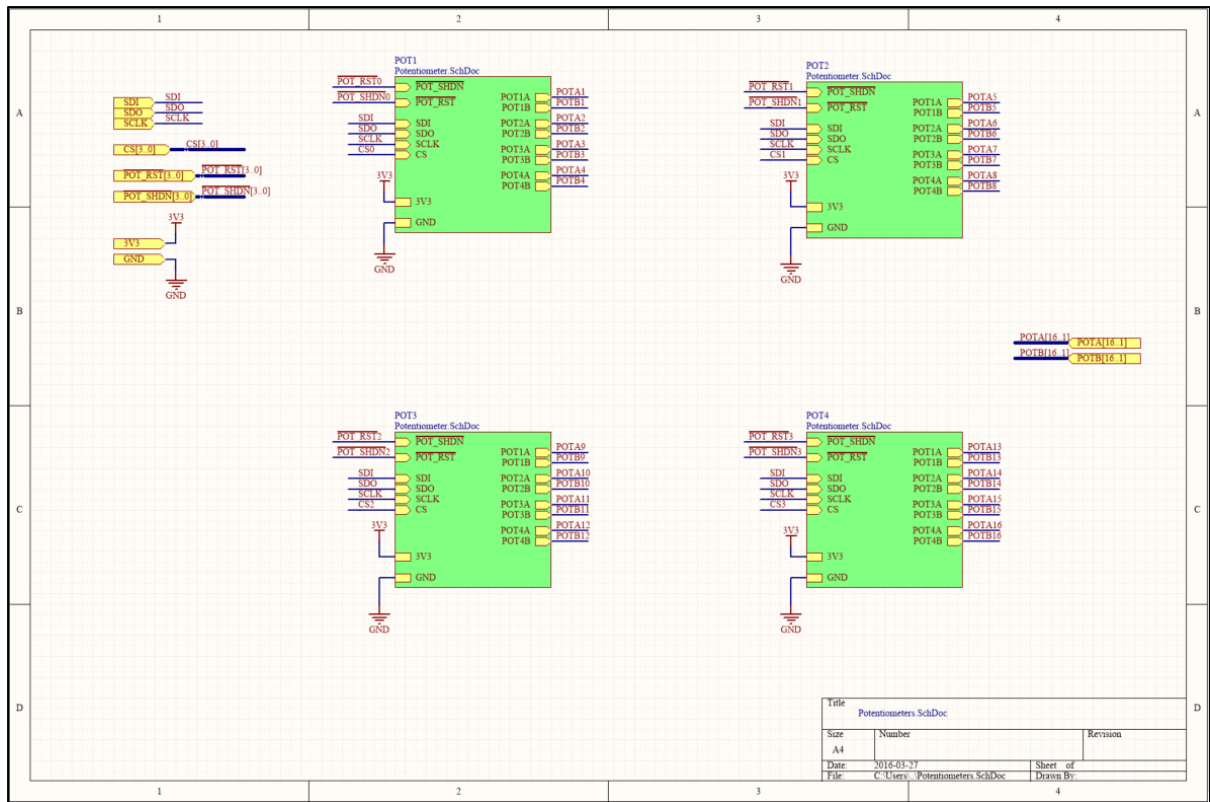


Figure-A IV-5 Schéma haut niveau des potentiomètres

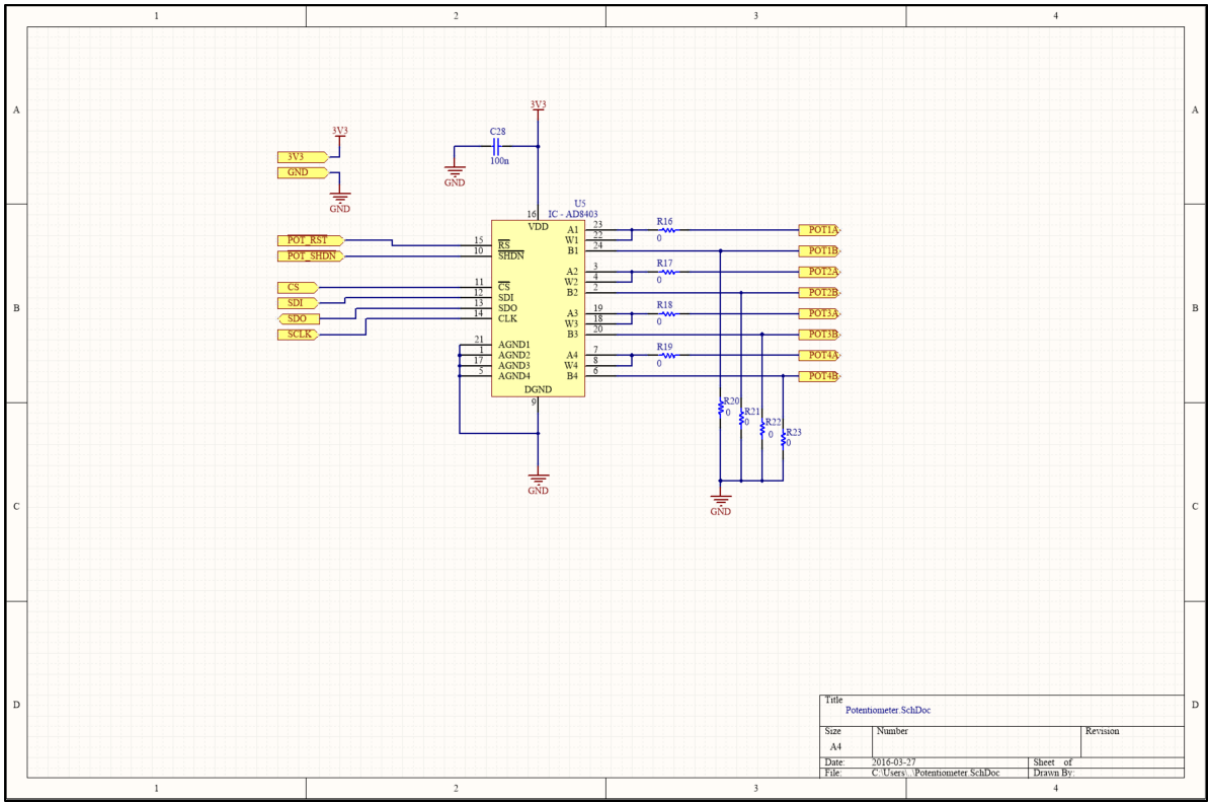


Figure-A IV-6 Circuit d'un potentiomètre

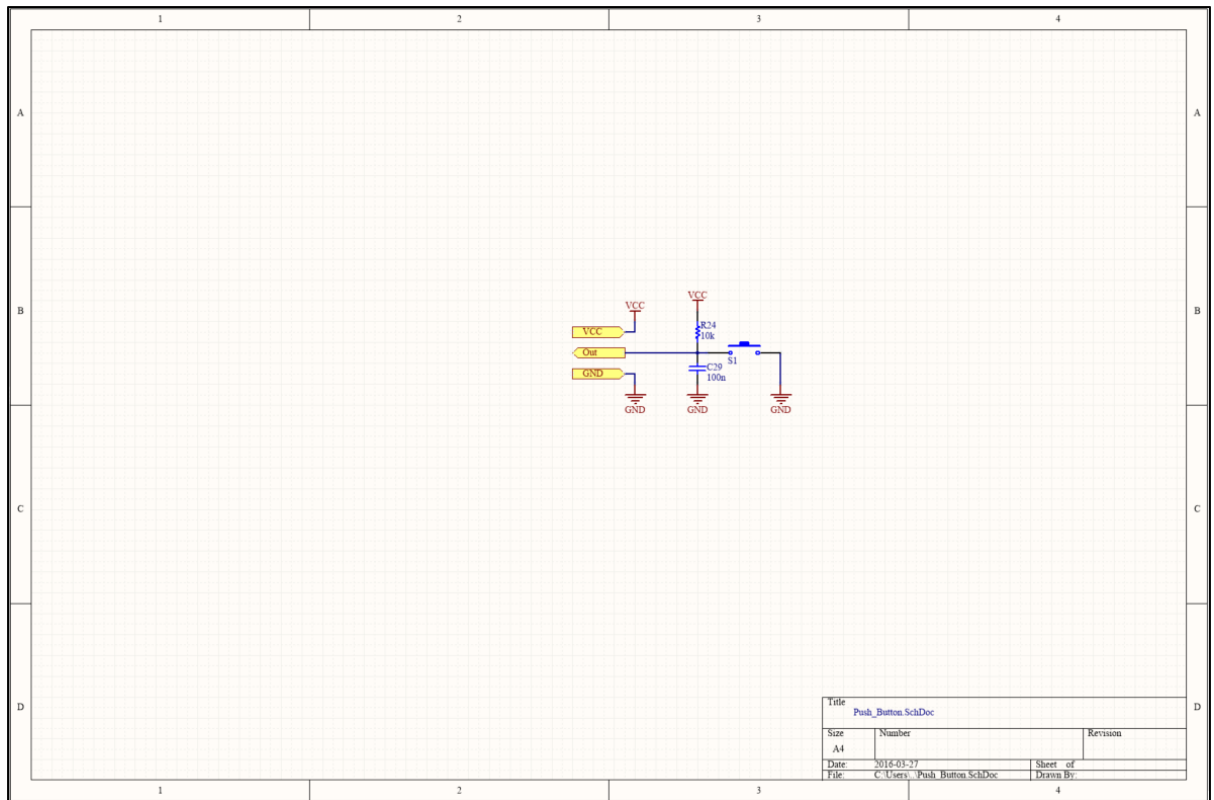


Figure-A IV-7 Circuit d'un interrupteur

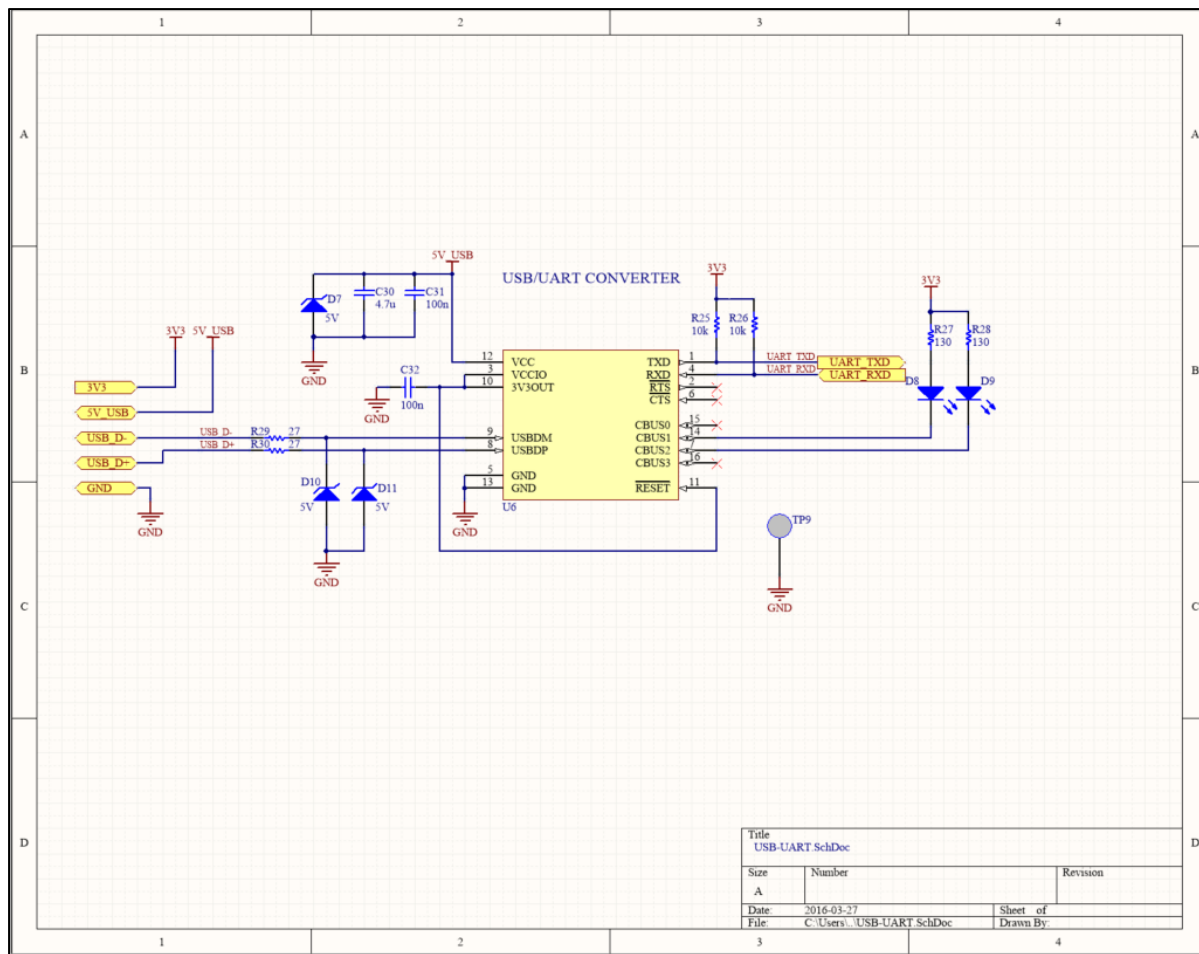


Figure-A IV-8 Circuit de conversion UART à USB

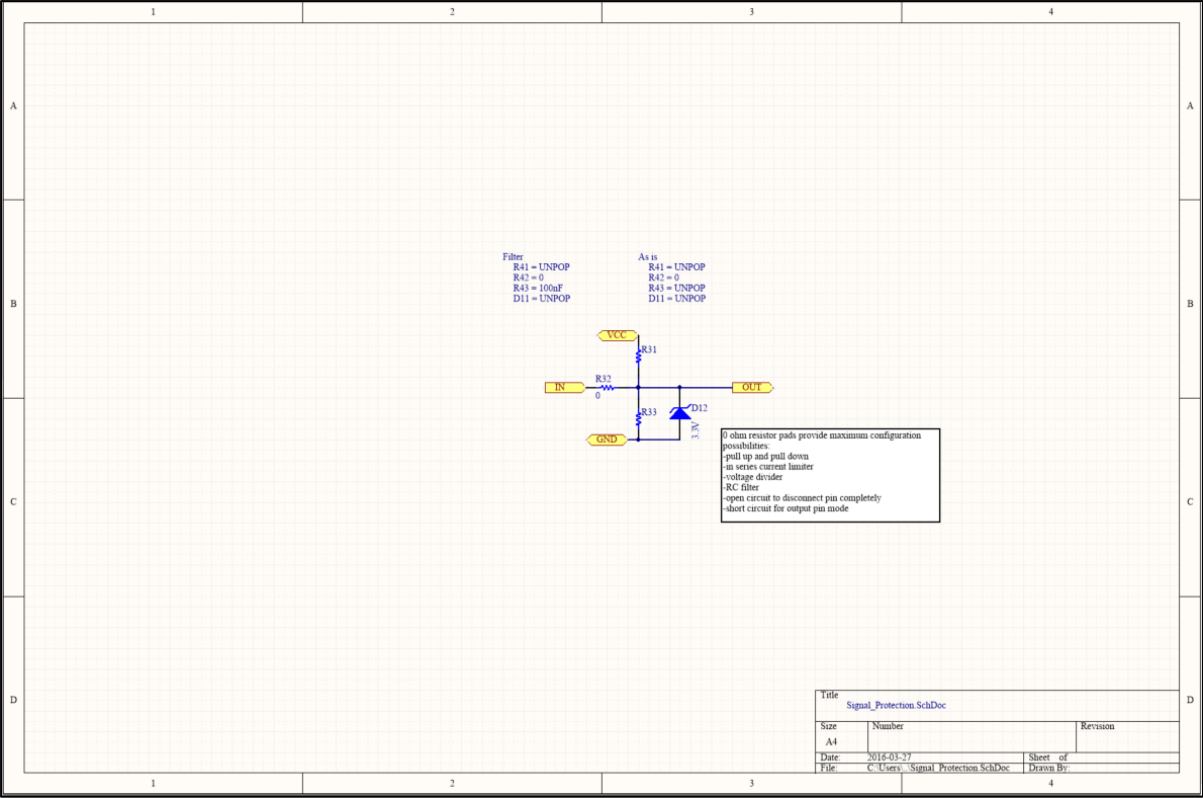


Figure-A IV-9 Circuit d'entrée du convertisseur analogue à numérique

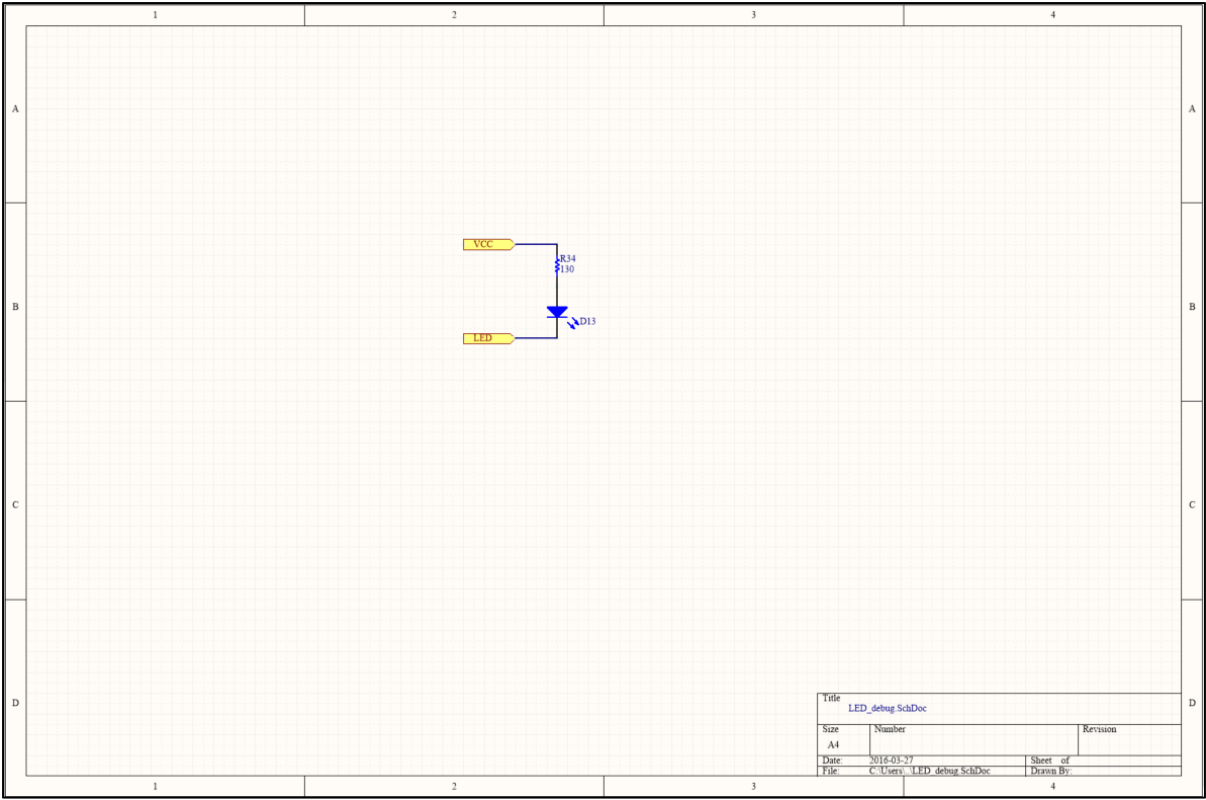


Figure-A IV-10 Circuit d'une DEL de débogage

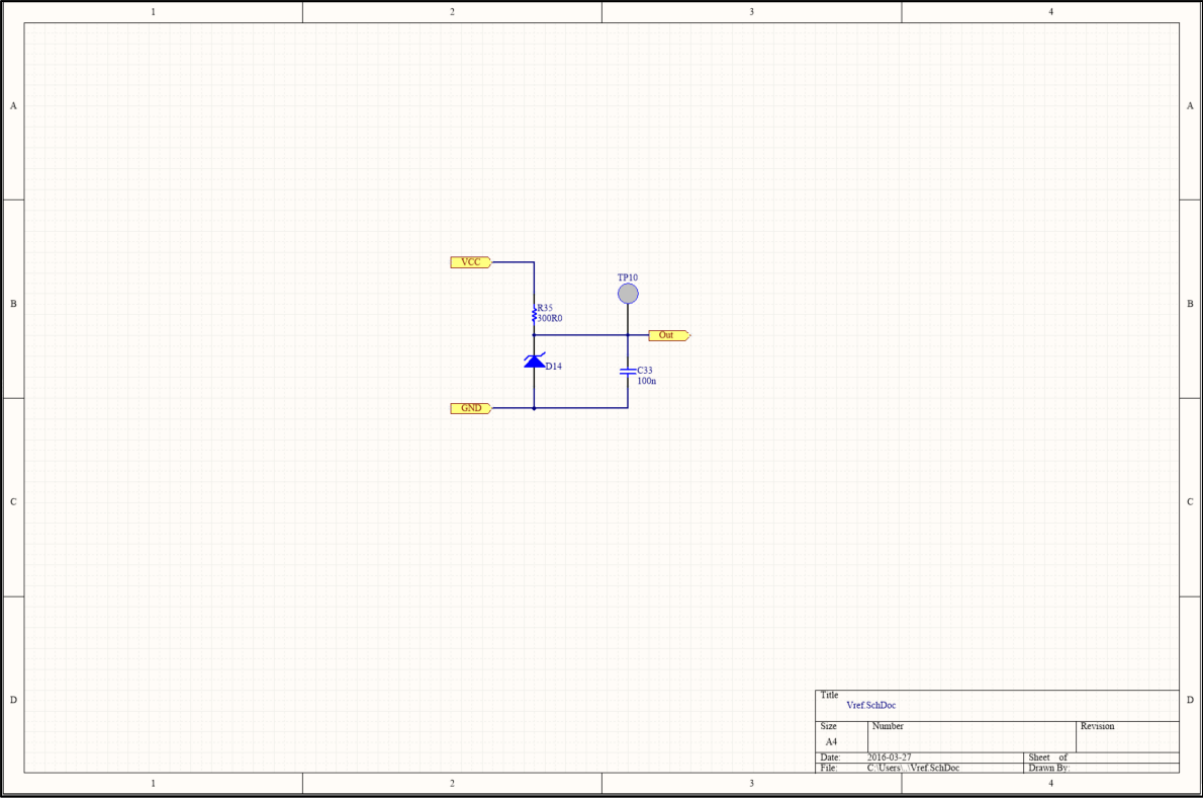


Figure-A IV-11 Circuit de tension de référence externe pour le convertisseur analogue à numérique

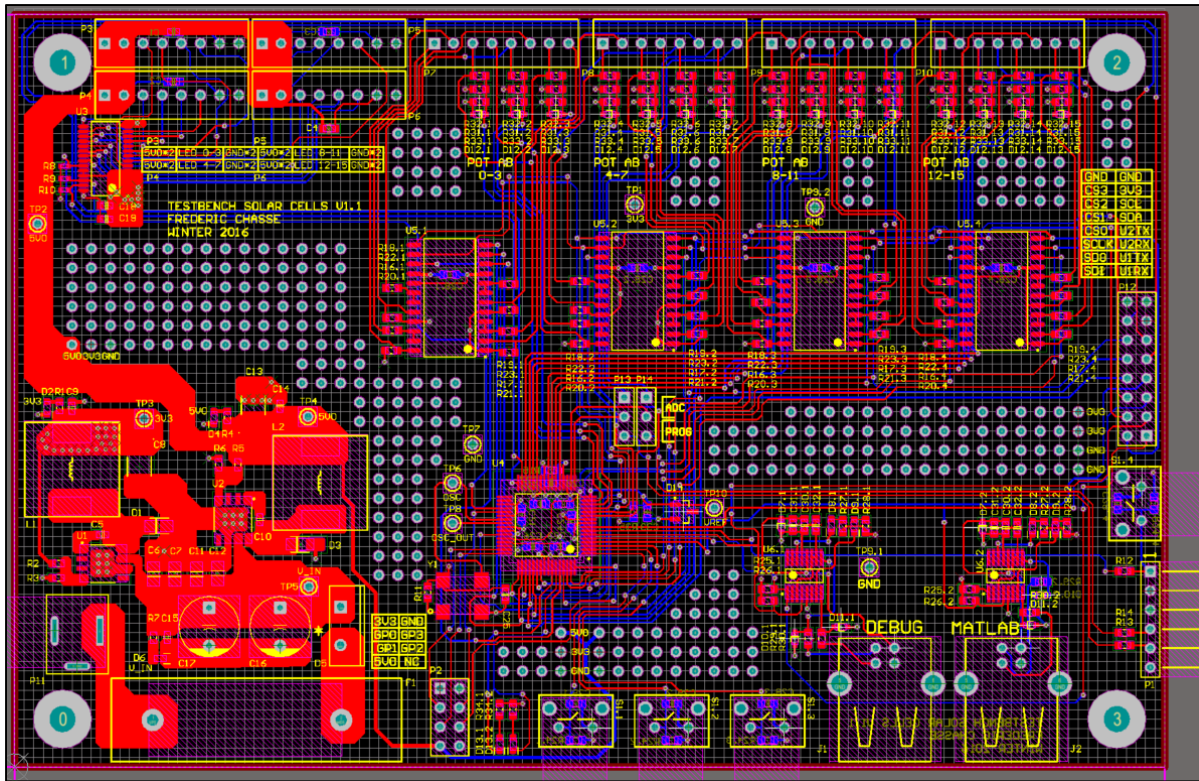


Figure-A IV-12 Dessin du PCB

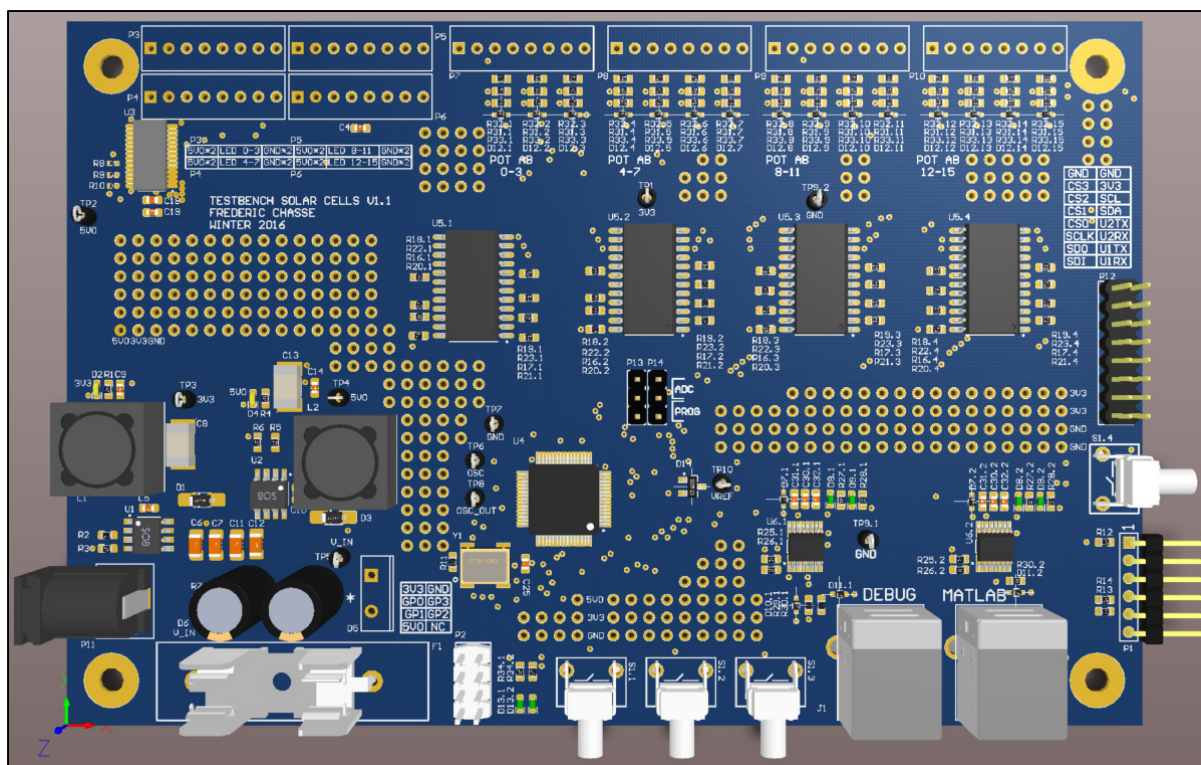


Figure-A IV-13 Modèle 3D du PCB

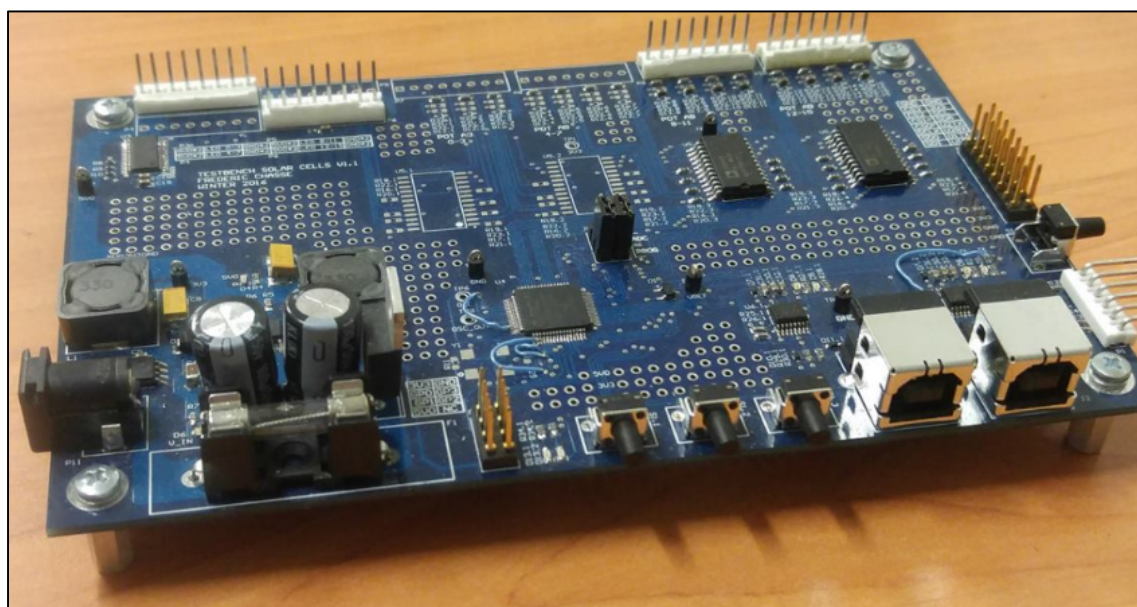


Figure-A IV-14 PCB partiellement assemblé

ANNEXE V

OPTIMISATION DE TROIS PCM PAR OEPS AVEC MAUVAISE DÉTECTION DE PERTURBATION PAR LES PARTICULES

La Figure-A V-1 montre un test préliminaire de l'application de l'OEPS sur 3 PCM lorsque la marge de détection d'une perturbation des sentinelles était de $\pm 5\%$. Comme on peut le voir dans la première sous-figure par les points indiqués, le R_{ext} de la PCM 3 n'a pas changé pendant deux itérations. Cependant, si l'on regarde les puissances à ces mêmes points, on voit que la PCM était encore en train de se stabiliser. On peut calculer la différence relative entre les deux points de la façon suivante :

$$Diff_{\%} = \frac{|P_2 - P_1|}{P_1} * 100\% = \frac{|0.001902 - 0.002005|}{0.002005} * 100\% = 5.14\%$$

Ainsi, avec une différence relative de 5.14%, l'OEPS a détecté une perturbation et a redémarré la recherche de l'optimum de la PCM 3. C'est donc pour cette raison que le paramètre de marge des sentinelles a été augmentée à $\pm 15\%$.

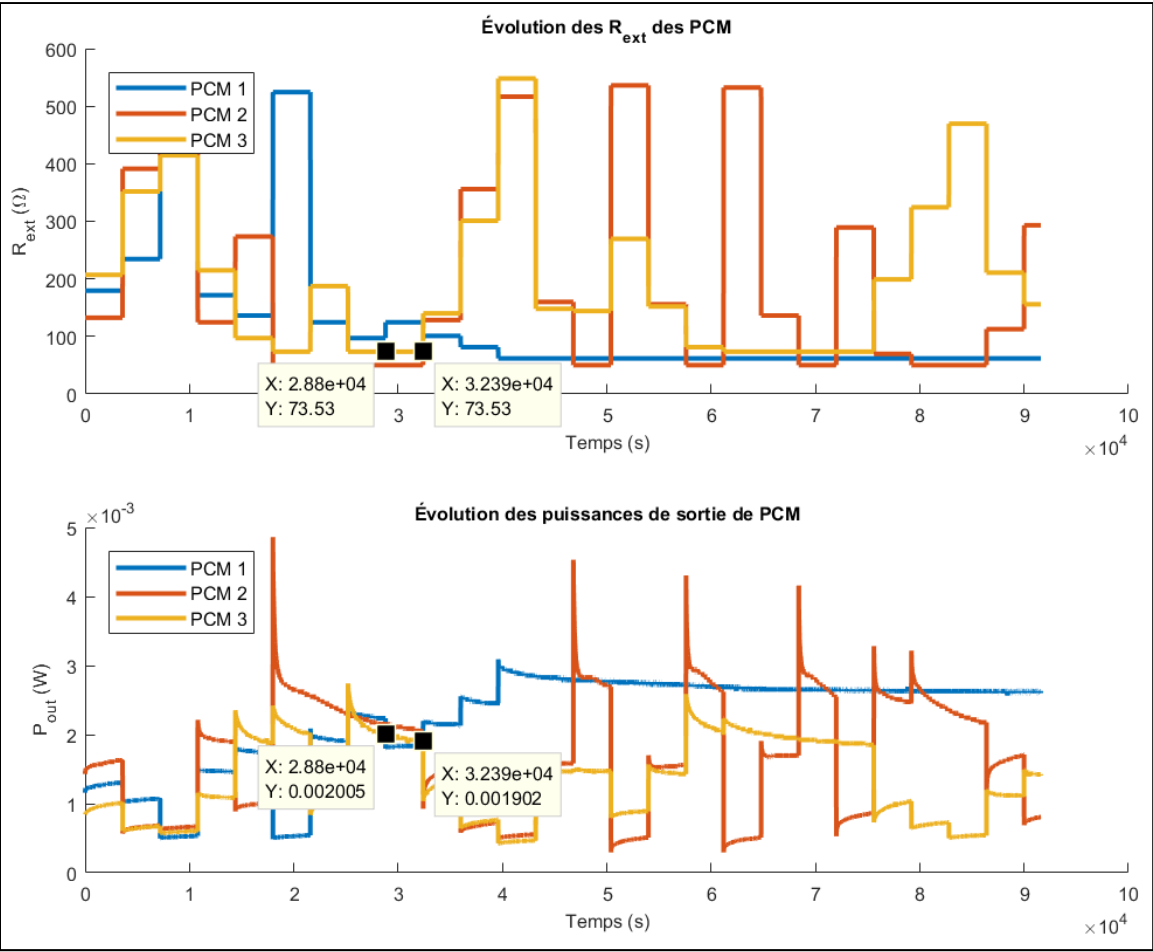


Figure-A V-1 Optimisation par OEPS de 3 PCM avec mauvaise détection de perturbation

ANNEXE VI

ANALYSE DE L'OEPPC AVEC AJOUT DE DIVERSITÉ POUR COMPENSER LES VARIATIONS TRÈS LENTES DE LA DYNAMIQUE INTERNE DES PCM

Afin d'adapter l'OEPPC au problème de dynamique lente des PCM observé pendant les expérimentations du CHAPITRE 3 (section 3.3), il a été proposé d'ajouter de la diversité à l'algorithme dans le but de suivre un optimum qui bouge lentement. Ainsi, une variable a été ajoutée à l'OEPPC, soit $T_{diversité}$. Cette variable dicte à l'algorithme après combien d'itérations en régime permanent une unité doit recommencer sa recherche d'optimum. Lors de ce redémarrage, le Δ est réinitialisé à Δ_1 . Lorsque la convergence est atteinte, Δ est fixé à Δ_2 puis, lorsque la convergence est à nouveau atteinte, on fixe l'unité à son D_{opt} . Ensuite, on compte le nombre d'itérations où l'unité reste en régime permanent, et le cycle recommence. Si une grande perturbation est détectée, l'unité est classifiée et placée dans un essaim OEPP.

Cette annexe sert donc à étudier en simulations les performances de l'OEPPC avec diversité (OEPPCD) dans un tel contexte. Le CHAPITRE 3 a déterminé que le P&O performe mieux que les algorithmes basés sur l'OEP pour ce type de problème, donc les performances de l'OEPPCD sont comparées à celles du P&O.

Pour ce faire, 10 PCM sont utilisées en simulation, dont le S_0 initial est de $S_0 = [600, 600, 550, 550, 500, 500, 575, 575, 525, 525] \text{ mg/L}$. À partir de l'itération 51, le S_0 de chaque PCM diminue de 0.5 mg/L à chaque itération, et ce jusqu'à l'itération 150, pour un S_0 final de $S_0 = [550, 550, 500, 500, 450, 450, 525, 525, 475, 475] \text{ mg/L}$. Les critères d'évaluation sont le temps de convergence, en nombre d'itérations, pour que les valeurs minimales et maximales de la variable de contrôle d'une unité ne varient que de $\pm 5\%$ (T_{conv}), la puissance moyenne en mW à partir de ce moment (P_{conv}), et l'énergie totale récoltée en joules (E_{total}). Pour tenir compte la nature stochastique de l'OEPPCD, 10 simulations seront faites dont les résultats moyennés sont présentés. Étant donné que le P&O est déterministe, les résultats d'une seule simulation sont donnés.

Le Tableau-A VI-1 donne les paramètres de P&O utilisés pour ces simulations, et le Tableau-A VI-2 donne les paramètres d'OEPPCD utilisés pour les simulations. La Figure-A VI-1 montre une simulation de l'optimisation par P&O de 10 PCM en présence de perturbations lentes, et la Figure-A VI-2 montre une simulation de l'optimisation par OEPPCD de 10 PCM dans ce même contexte.

On voit sur les figures que les deux algorithmes réussissent bien à suivre les optima changeants des unités. On observe aussi que l'OEPPCD présente beaucoup moins d'oscillations que le P&O

Tableau-A VI-1 Paramètres de P&O utilisés pour les simulations

Paramètre	Valeur	Paramètre	Valeur
Δ	15 Ω	Position minimale	10 Ω
Position initiale	200 Ω	Position maximale	350 Ω

Tableau-A VI-2 Paramètres d'OEPPCD utilisés pour les simulations

Paramètre	Valeur	Paramètre	Valeur
c_1	0.5	Δ_c	$\pm 10 \Omega$
c_2	1.2	Position minimale	10 Ω
ω	0.3	Position maximale	350 Ω
Nombre minimal de particules	3	Δ_1 (P&O)	5 Ω
Marge des sentinelles	$\pm 3\%$	Δ_2 (P&O)	1 Ω
Nombre d'itérations définissant la convergence pour l'OEPP	5	Amplitude de convergence pour l'OEPP	$\pm 1\%$
Nombre d'itérations définissant la convergence pour le P&O	7	Amplitude de convergence pour le P&O	$\pm 2\Delta$
$T_{diversité}$	15		

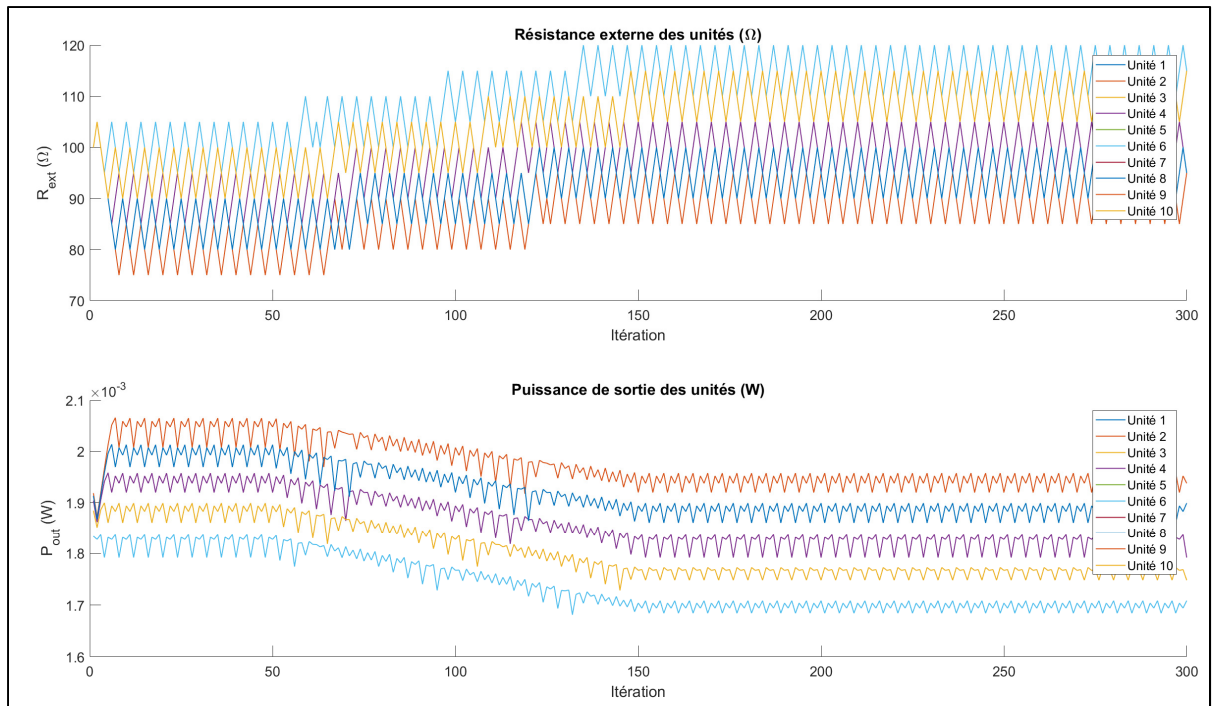


Figure-A VI-1 Optimisation par P&O de 10 PCM avec perturbations lentes

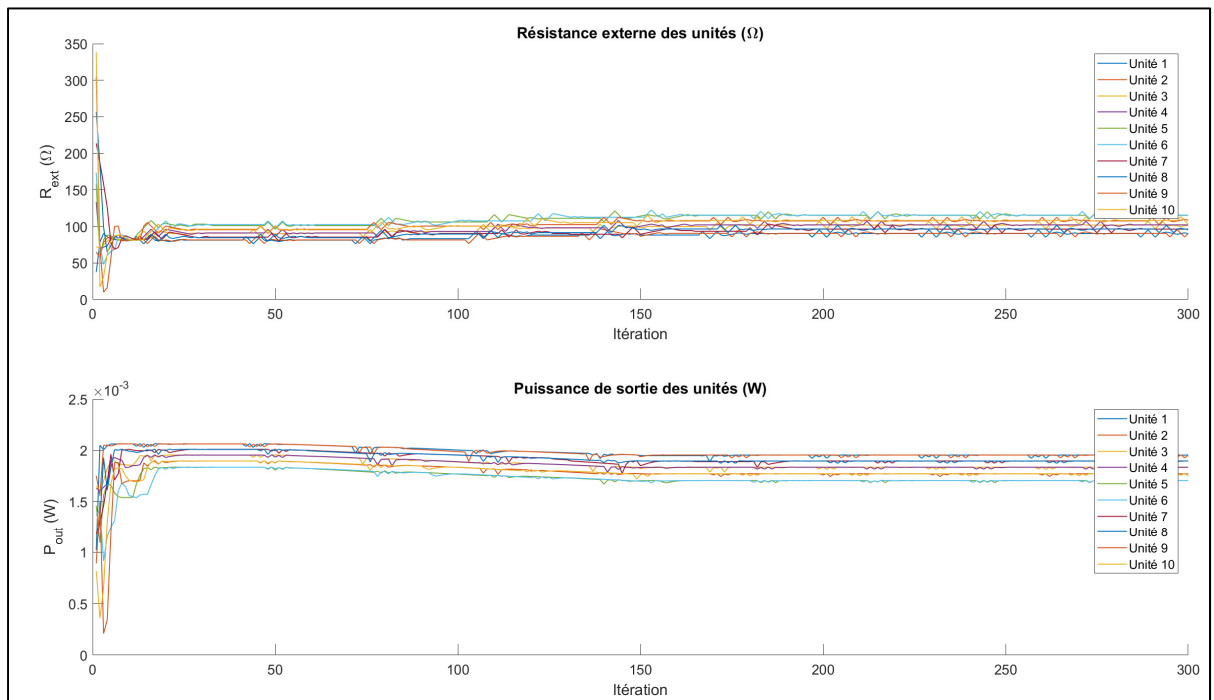


Figure-A VI-2 Optimisation par OEPPCD de 10 PCM en présence de perturbations lentes

Le Tableau-A VI-3 donne les résultats de l'optimisation avec ces deux algorithmes. Il est à noter que le temps de convergence très long de l'OEPPCD ne reflète pas nécessairement la réalité, car le calcul de T_{conv} est fait selon la variation de l'amplitude des sommets de l'oscillation de la variable de commande d'une unité. Or, dans notre cas, en raison de la variable $T_{diversité}$ ajoutée, les unités de l'OEPPCD sont très stables pendant 15 itérations, puis recommencent à varier selon Δ_1 et Δ_2 . On voit cependant qu'en terme d'énergie récoltée, l'OEPPCD et le P&O sont à peu près semblables. Par contre, étant donné le nombre moindre d'oscillations de l'OEPPCD par rapport au P&O, l'OEPPCD réussit à obtenir une puissance moyenne finale plus élevée que le P&O. Ainsi, l'OEPPCD performe mieux que le P&O dans un contexte d'optimum qui varie lentement.

Tableau-A VI-3 Résultats de simulation de 10 PCM avec perturbations lentes

Algorithme	E_{total} (joules)	T_{conv} (itérations)	P_{conv} (mW)
P&O	48044.404	128.2	18.241
OEPPCD	48049.056	286.49	18.319

Il est aussi intéressant de comparer les performances des deux algorithmes dans un contexte où les PCM asservies ne sont pas sujettes à des perturbations. Ainsi, des simulations ont été faites à cet égard. La Figure-A VI-3 montre une simulation de l'optimisation par P&O de 10 PCM sans perturbations, et la Figure-A VI-4 montre une simulation de l'optimisation par OEPPCD pour ce même cas d'étude.

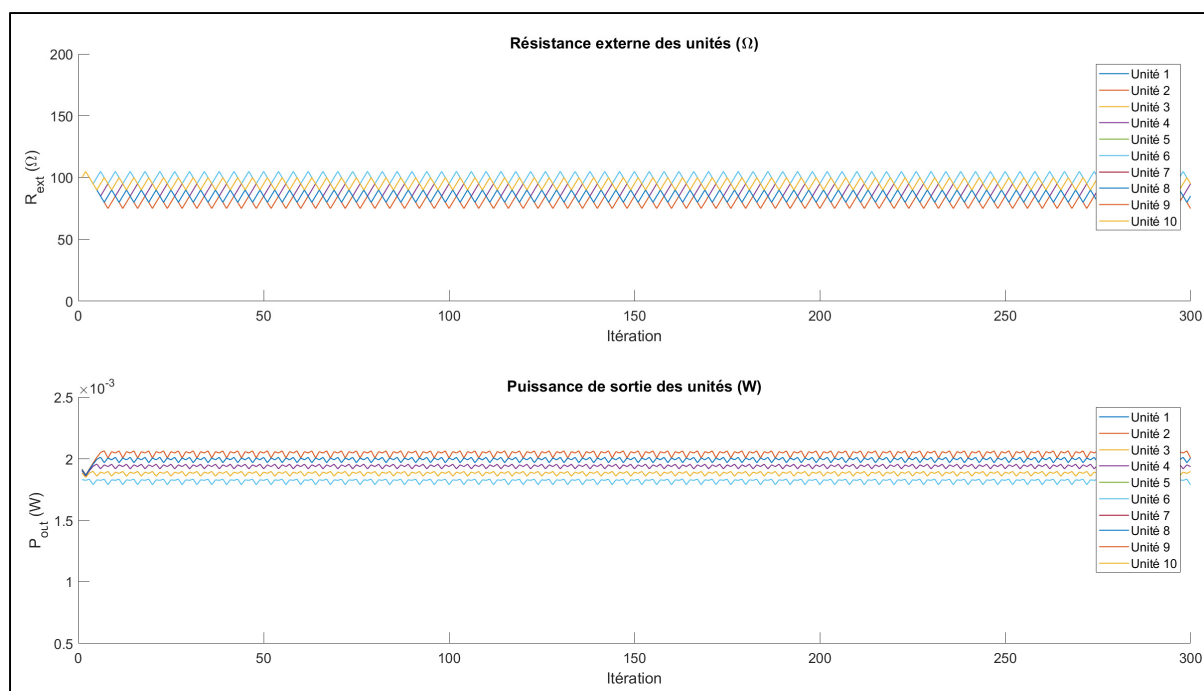


Figure-A VI-3 Optimisation par P&O de 10 PCM sans perturbations

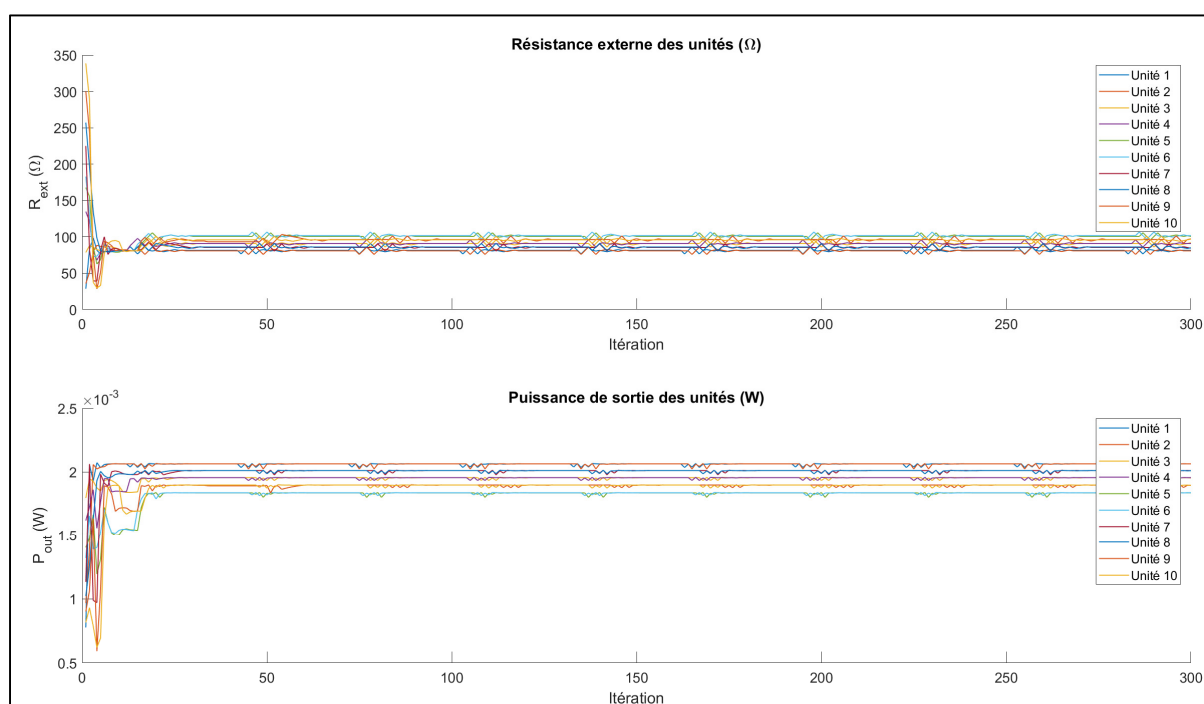


Figure-A VI-4 Optimisation par OEPPCD de 10 PCM sans perturbations

On voit que, bien que l'OEPPCD présente un certain niveau d'oscillations en régime permanent, ce niveau est quand même moindre que celui du P&O. Le Tableau-A VI-4 donne les résultats des simulations pour ces deux algorithmes. L'OEPPCD réussit à obtenir une puissance moyenne finale plus haute que le P&O, de même qu'une quantité d'énergie récoltée plus élevée. Ainsi, nos simulations indiquent que l'OEPPCD pourrait potentiellement avoir une meilleure performance que le P&O pour optimiser des PCM dont la dynamique interne change très lentement, tout comme des PCM dont la dynamique interne est stable.

Tableau-A VI-4 Résultats de simulation de 10 PCM sans perturbations

Algorithme	E_{total} (joules)	T_{conv} (itérations)	P_{conv} (mW)
P&O	50040.229	4.6	19.376
OEPPCD	50128.125	293.34	19.505

ANNEXE VII

SCHÉMA ÉLECTRIQUE DU PCB DE CELLULES PV

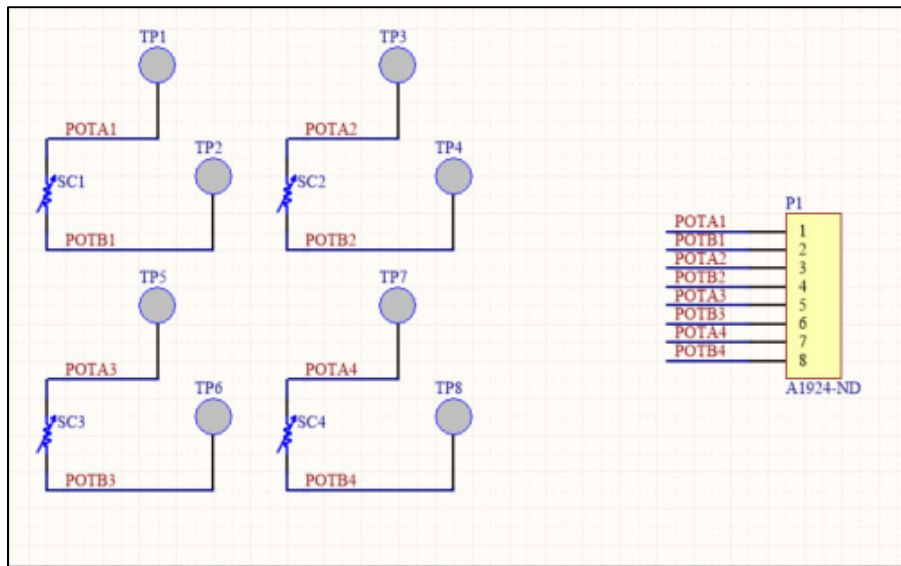


Figure-A VII-1 Schéma électrique

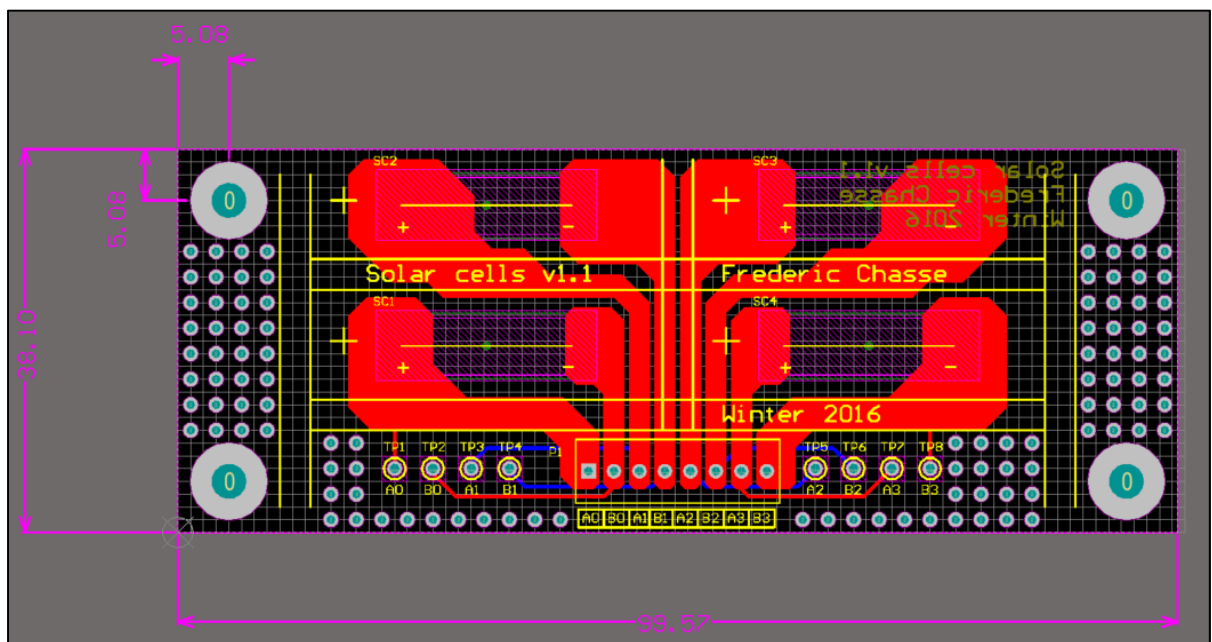


Figure-A VII-2 Dessin du PCB

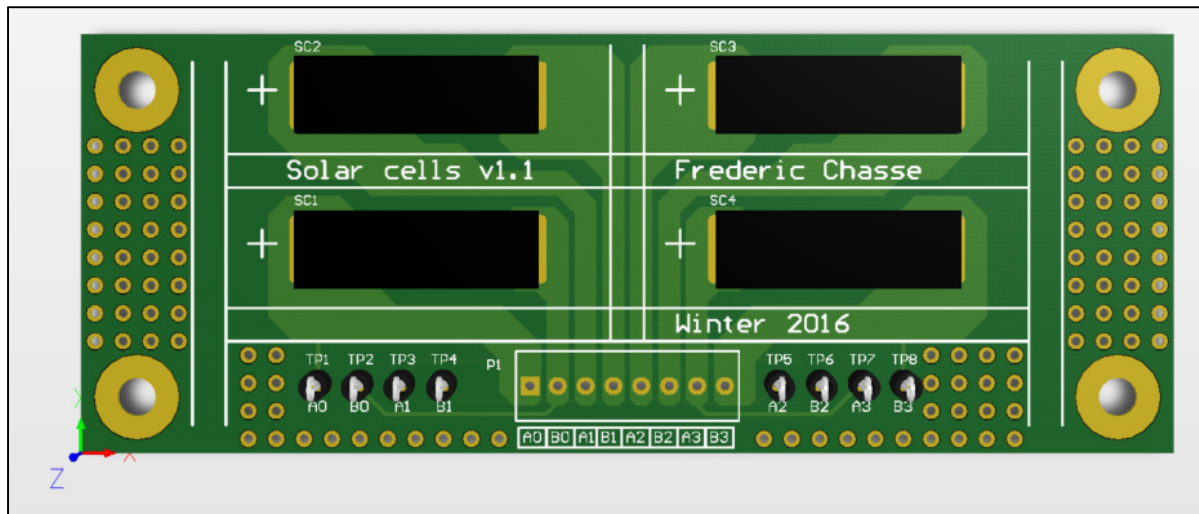


Figure-A VII-3 Modèle 3D du PCB

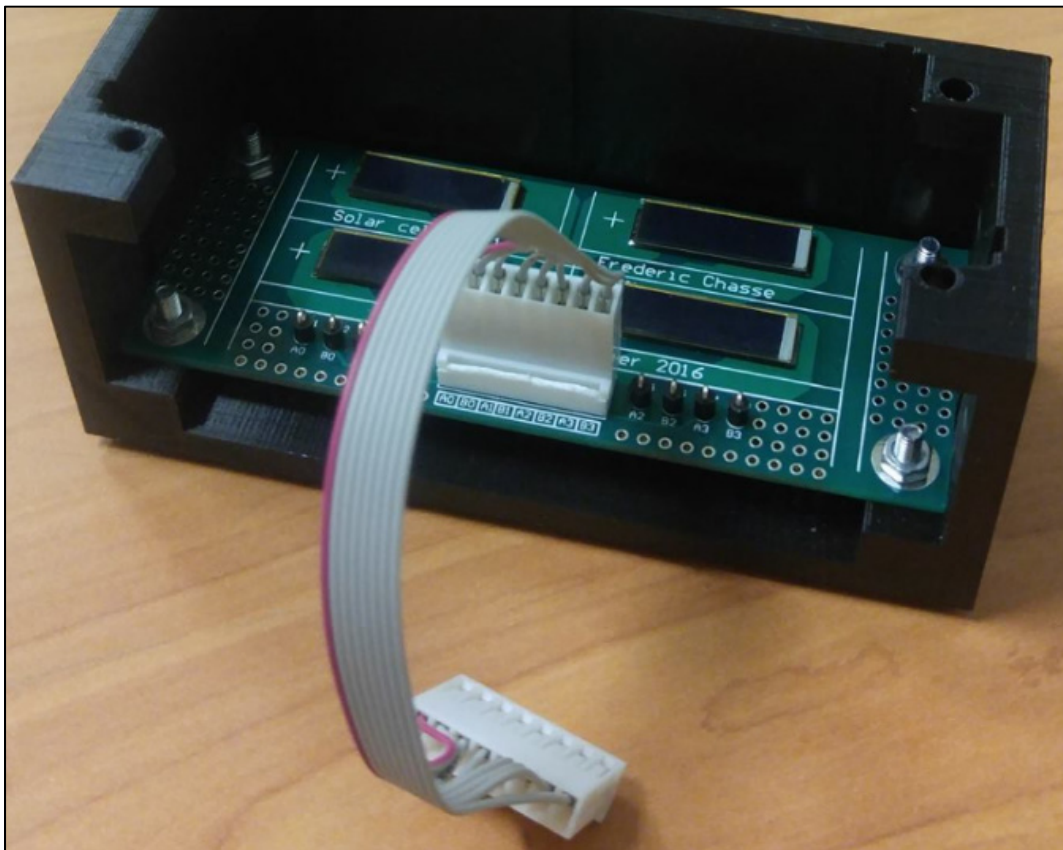


Figure-A VII-4 Photo du PCB assemblé et monté dans son boîtier

ANNEXE VIII

SCHÉMA ÉLECTRIQUE DU CIRCUIT DE DEL POUR LES CELLULES PV

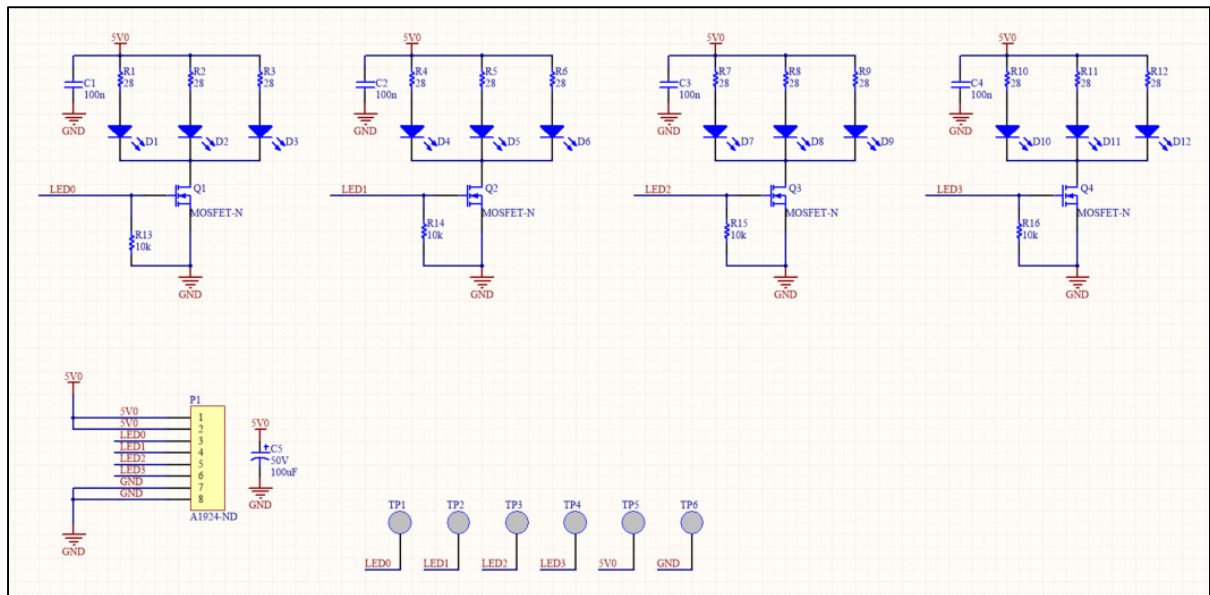


Figure-A VIII-1 Circuit électrique

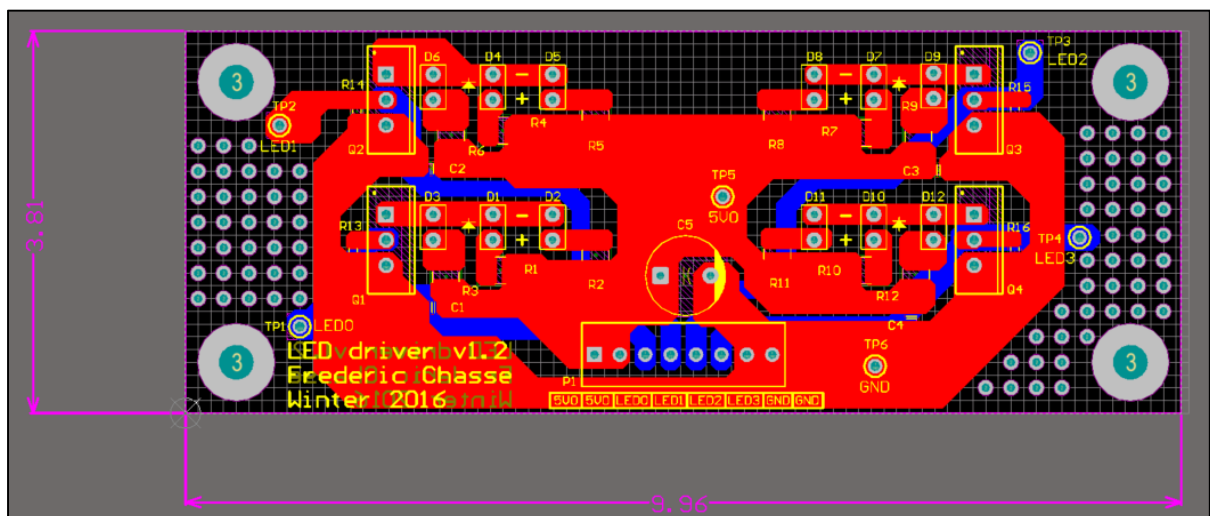


Figure-A VIII-2 Dessin du PCB

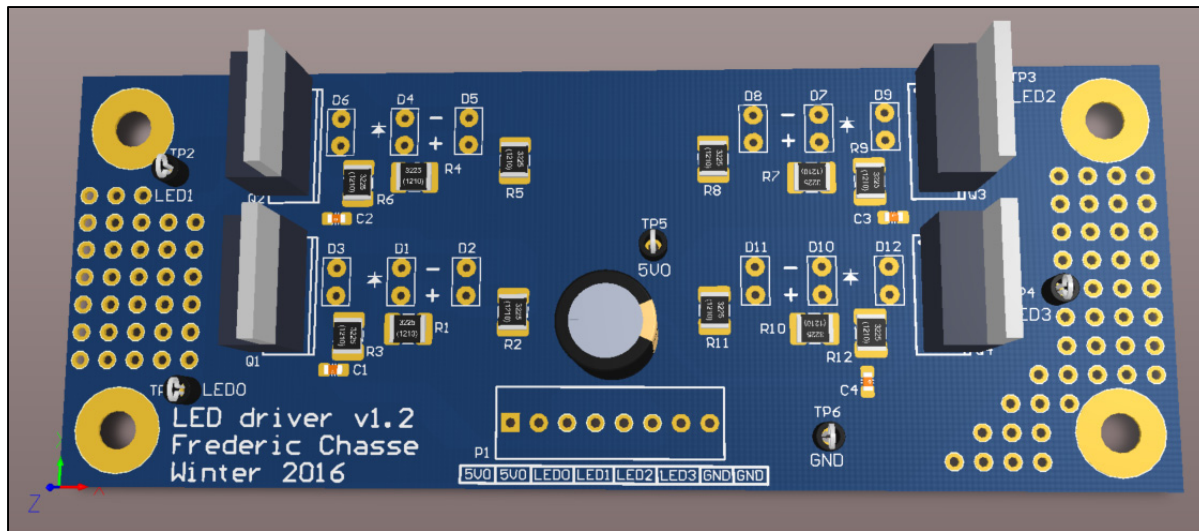


Figure-A VIII-3 Modèle 3D du PCB

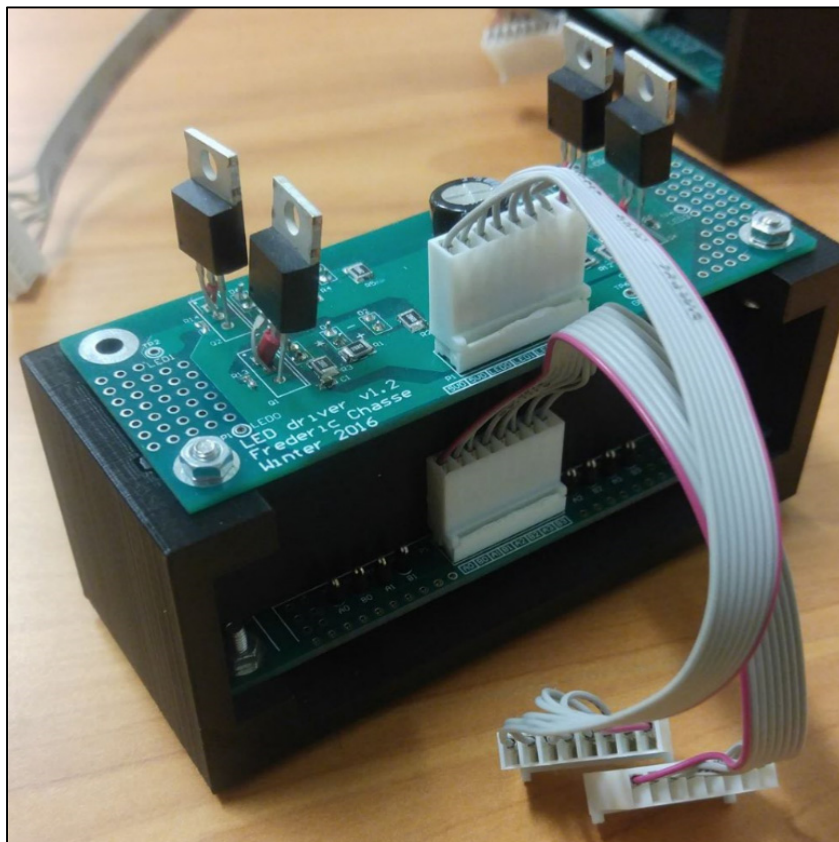


Figure-A VIII-4 Vue du dessus du PCB

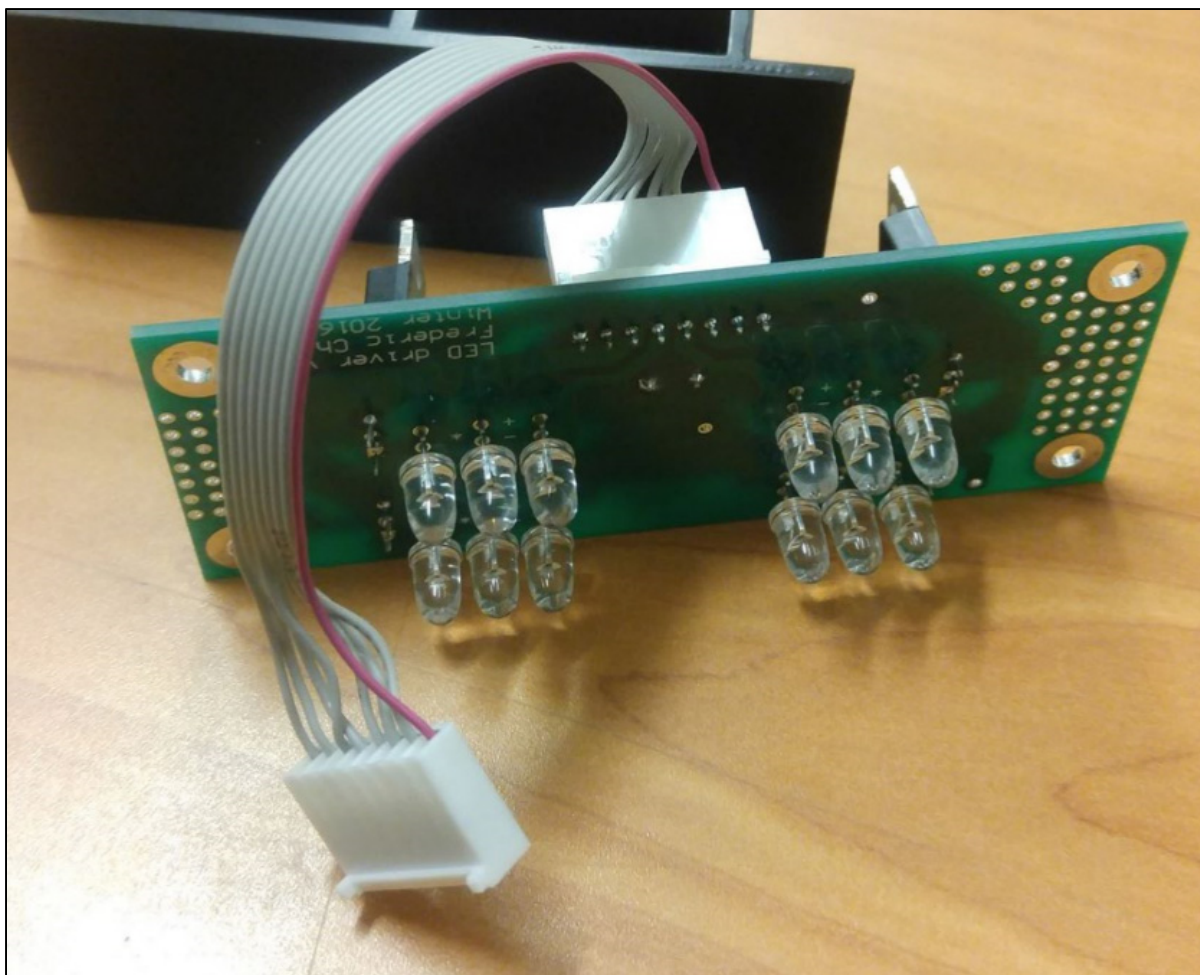


Figure-A VIII-5 Vue du dessous du PCB

ANNEXE IX

DESSINS MÉCANIQUES DU BOÎTIER DES CELLULES PV

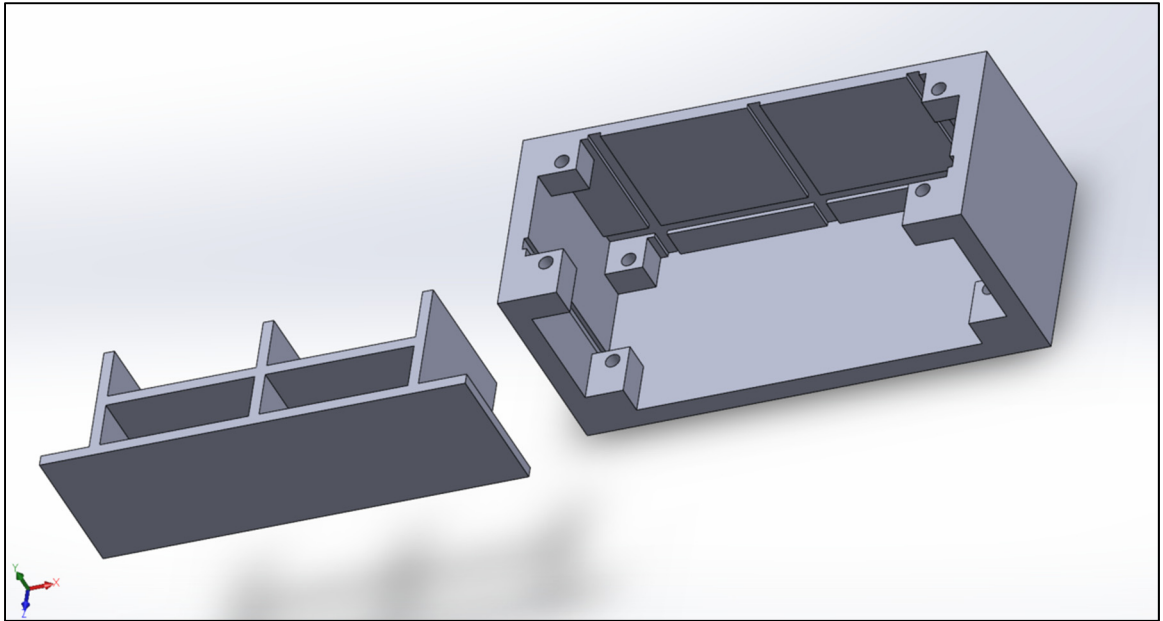


Figure-A IX-1 Modèle 3D du boîtier et du mur de séparation des cellules PV

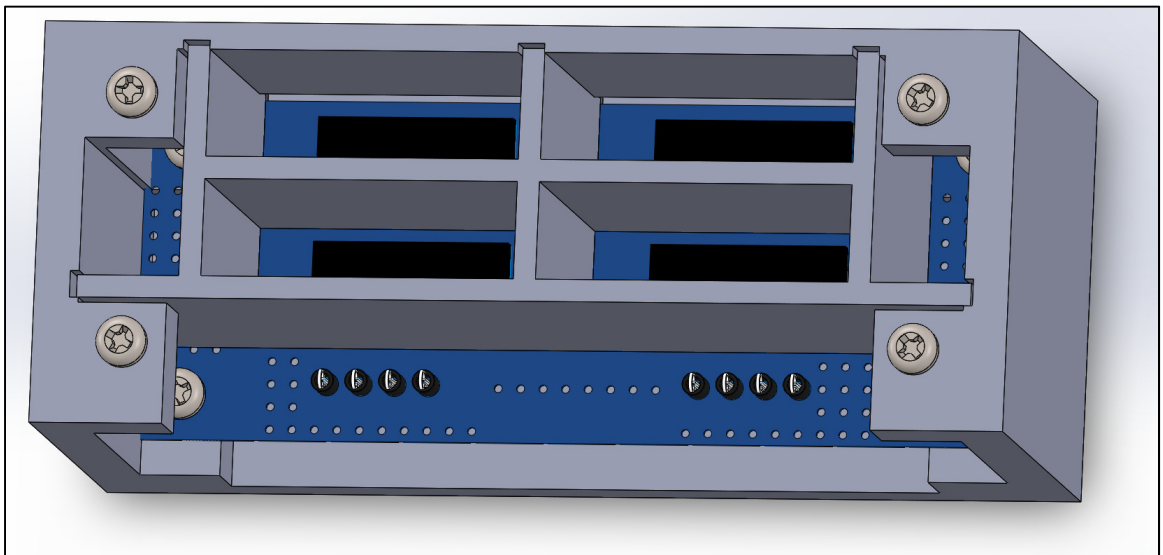


Figure-A IX-2 Assemblage 3D du PCB de cellules PV, du boîtier, et du mur de séparation

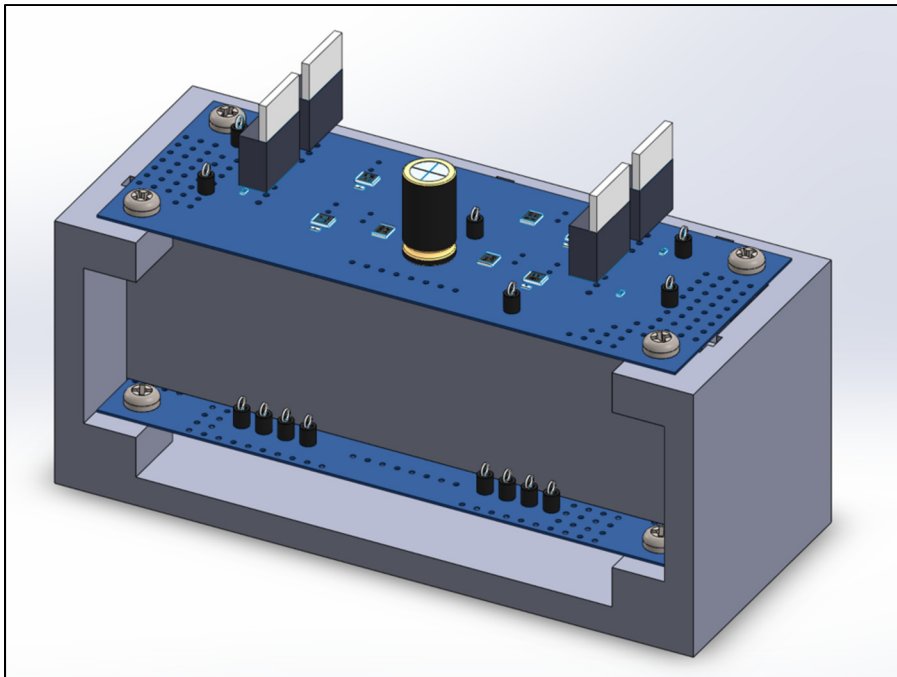


Figure-A IX-3 Assemblage 3D du boîtier, du mur de séparation, et des PCB de DEL et de cellules PV

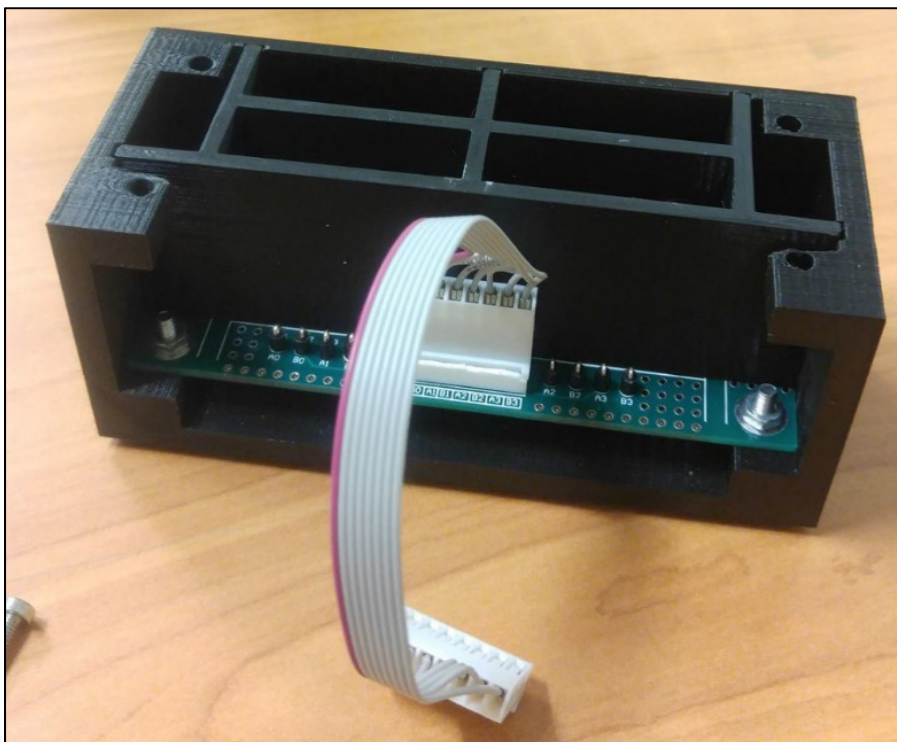


Figure-A IX-4 Assemblage du boîtier et du PCB de cellules PV

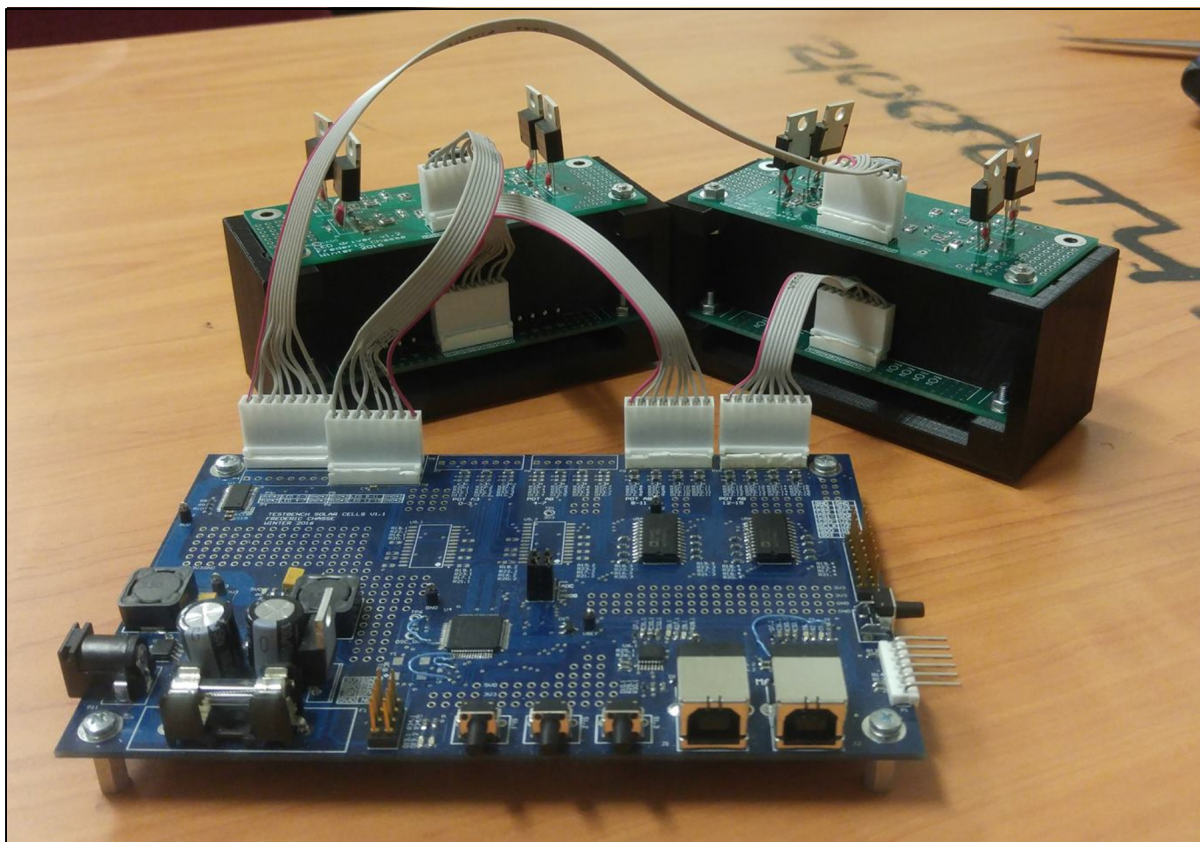


Figure-A IX-5 Montage mécanique du banc d'essai de cellules PV

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Azzouzi, Messaouda. 2012. « Comparaison between MPPT P&O and MPPT fuzzy controls in optimizing the photovoltaic generator ». *International Journal of Advanced Computer Science and Applications*, vol. 3, n° 12, p. 57-62.
- Carrette, L, KA Friedrich et U1 Stimming. 2001. « Fuel cells–fundamentals and applications ». *Fuel cells*, vol. 1, n° 1, p. 5-39.
- Chassé, Frédéric, et Lyne Woodward. 2016. « A New Approach to Particle Swarm Optimization for Dynamic Systems With Multiple Units ». In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. p. 4031-4036. IEEE.
- Chen, Wei-Ru, Liang-Rui Chen, Chia-Hsuan Wu et Ci-Min Lai. 2015. « Multiclusterc-based particle swarm optimization algorithm for photovoltaic maximum power point tracking ». In *Future Energy Electronics Conference (IFEEC), 2015 IEEE 2nd International*. p. 1-6. IEEE.
- Du, Zhuwei, Haoran Li et Tingyue Gu. 2007. « A state of the art review on microbial fuel cells: A promising technology for wastewater treatment and bioenergy ». *Biotechnology Advances*, vol. 25, n° 5, p. 464-482.
- Eberhart, Russ C, et James Kennedy. 1995. « A new optimizer using particle swarm theory ». In *Proceedings of the sixth international symposium on micro machine and human science*. Vol. 1, p. 39-43. New York, NY.
- Guillemette, Jean-Sébastien, et Lyne Woodward. 2014. « Maximizing wind farm energy production in presence of aerodynamic interactions ». In *International Conference of Control, Dynamic Systems and Robotics*.
- Hu, Xiaohui, et Russell C Eberhart. 2002. « Adaptive particle swarm optimization: detection and response to dynamic systems ». In *wcci*. p. 1666-1670. IEEE.
- Huynh, Duy C, Thoai N Nguyen, Matthew W Dunnigan et Markus A Mueller. 2013. « Dynamic particle swarm optimization algorithm based maximum power point tracking of solar photovoltaic panels ». In *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*. p. 1-6. IEEE.
- Ishaque, Kashif, Zainal Salam, Muhammad Amjad et Saad Mekhilef. 2012. « An improved Particle Swarm Optimization (PSO)–based MPPT for PV with reduced steady-state oscillation ». *Power Electronics, IEEE Transactions on*, vol. 27, n° 8, p. 3627-3638.

- Kebir, A., L. Woodward et O. Akhrif. 2018. « Extremum-Seeking Control With Adaptive Excitation: Application to a Photovoltaic System ». *IEEE Transactions on Industrial Electronics*, vol. 65, n° 3, p. 2507-2517.
- Khaehintung, Noppadol, Theerayod Wiangtong et Phaophak Sirisuk. 2006. « FPGA implementation of MPPT using variable step-size P&O algorithm for PV applications ». In *Communications and Information Technologies, 2006. ISCIT'06. International Symposium on*. p. 212-215. IEEE.
- Leyva, Ramon, Corinne Alonso, Isabelle Queinnec, Angel Cid-Pastor, Denis Lagrange et Luis Martínez-Salamero. 2006. « MPPT of photovoltaic systems using extremum-seeking control ». *IEEE transactions on aerospace and electronic systems*, vol. 42, n° 1, p. 249-258.
- Li, Wen-Wei, Han-Qing Yu et Zhen He. 2014. « Towards sustainable wastewater treatment by using microbial fuel cells-centered technologies ». *Energy & Environmental Science*, vol. 7, n° 3, p. 911-924.
- Li, Xiao, Yaoyu Li, John E Seem et Peng Lei. 2011. « Maximum power point tracking for photovoltaic systems using adaptive extremum seeking control ». In *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*. p. 803-810. American Society of Mechanical Engineers.
- Liu, Xuejun, et Luiz AC Lopes. 2004. « An improved perturbation and observation maximum power point tracking algorithm for PV arrays ». In *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*. Vol. 3, p. 2005-2010. IEEE.
- Logan, Bruce E. 2008. *Microbial fuel cells*. John Wiley & Sons.
- Matsumoto, Makoto, et Takuji Nishimura. 1998. « Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator ». *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, n° 1, p. 3-30.
- Mehenna, Toufik, et Lyne Woodward. 2012. « Maximizing wind energy production with the multi-unit optimization method ». In *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. p. 1-6. IEEE.
- Melhuish, Chris, Ioannis Ieropoulos, John Greenman et Ian Horsfield. 2006. « Energetically autonomous robots: food for thought ». *Autonomous Robots*, vol. 21, n° 3, p. 187-198.
- Molognoni, Daniele, Sebastià Puig, M Dolors Balaguer, Alessandro Liberale, Andrea G Capodaglio, Arianna Callegari et Jesús Colprim. 2014. « Reducing start-up time and minimizing energy losses of Microbial Fuel Cells using Maximum Power Point Tracking strategy ». *Journal of Power Sources*, vol. 269, p. 403-411.

- Nguyen, Trung Thanh, Shengxiang Yang et Juergen Branke. 2012. « Evolutionary dynamic optimization: A survey of the state of the art ». *Swarm and Evolutionary Computation*, vol. 6, p. 1-24.
- O'Neill, Melissa E. 2014. *PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation* (September 2014). Technical report. Harvey Mudd College, 46 p.
- Pinto, RP, B Srinivasan, SR Guiot et B Tartakovsky. 2011. « The effect of real-time external resistance optimization on microbial fuel cell performance ». *Water research*, vol. 45, n° 4, p. 1571-1578.
- Pinto, RP, B Srinivasan, M-F Manuel et B Tartakovsky. 2010. « A two-population bio-electrochemical model of a microbial fuel cell ». *Bioresource technology*, vol. 101, n° 14, p. 5256-5265.
- Potter, Michael C. 1911. « Electrical effects accompanying the decomposition of organic compounds ». *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, p. 260-276.
- Radziemska, E. 2003. « The effect of temperature on the power drop in crystalline silicon solar cells ». *Renewable Energy*, vol. 28, n° 1, p. 1-12.
- Ratnaweera, Asanga, Saman K Halgamuge et Harry C Watson. 2004. « Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients ». *Evolutionary Computation, IEEE Transactions on*, vol. 8, n° 3, p. 240-255.
- Recio-Garrido, Dídac, Michel Perrier et Boris Tartakovsky. 2016. « Combined bioelectrochemical–electrical model of a microbial fuel cell ». *Bioprocess and Biosystems Engineering*, vol. 39, n° 2, p. 267-276.
- Shantaram, Avinash, Haluk Beyenal, Raaja Raajan Angathevar Veluchamy et Zbigniew Lewandowski. 2005. « Wireless sensors powered by microbial fuel cells ». *Environmental science & technology*, vol. 39, n° 13, p. 5037-5042.
- Sridhar, Rr, N Thamizh Selvan, S Jeevananthan et PV Sujith Chowdary. 2010. « Performance improvement of a photo voltaic array using MPPT (P&O) technique ». In *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*. p. 191-195. IEEE.
- Srinivasan, B. 2007. « Real-time optimization of dynamic systems using multiple units ». *International Journal of Robust and Nonlinear Control*, vol. 17, n° 13, p. 1183-1193.

- Sundareswaran, K., V. Vignesh kumar et S. Palani. 2015. « Application of a combined particle swarm optimization and perturb and observe method for MPPT in PV systems under partial shading conditions ». *Renewable Energy*, vol. 75, n° Supplement C, p. 308-317.
- Wang, Heming, Jae-Do Park et Zhiyong Ren. 2012. « Active energy harvesting from microbial fuel cells at the maximum power point without using resistors ». *Environmental science & technology*, vol. 46, n° 9, p. 5247-5252.
- Wang, Heming, Jae-Do Park et Zhiyong Jason Ren. 2015. « Practical Energy Harvesting for Microbial Fuel Cells: A Review ». *Environmental science & technology*, vol. 49, n° 6, p. 3267-3277.
- Woodward, L, M Perrier, B Srinivasan, RP Pinto et B Tartakovsky. 2010. « Comparison of real-time methods for maximizing power output in microbial fuel cells ». *AIChE Journal*, vol. 56, n° 10, p. 2742-2750.
- Woodward, Lyne. 2009. *Adaptation de la méthode multi-unités à l'optimisation sous contraintes en présence d'unités non identiques*, 70. Mémoire de thèse doctorale. 12. École Polytechnique de Montréal, 199 p.
- Woodward, Lyne, Boris Tartakovsky, Michel Perrier et Bala Srinivasan. 2009. « Maximizing power production in a stack of microbial fuel cells using multiunit optimization method ». *Biotechnology progress*, vol. 25, n° 3, p. 676-682.

